



NATIONAL OPEN UNIVERSITY OF NIGERIA

SCHOOL OF SCIENCE AND TECHNOLOGY

COURSE CODE: CIT 322

COURSE TITLE: INTRODUCTION TO INTERNET PROGRAMMING

CIT 322: Introduction to Internet Programming (3 units)

Introduction to current programming models in generating and supporting rich real-world web based applications. Internet architecture and organization.

Internet services, electronic mail, data transfer, dial-up, connection protocols.

Connection to Internet: modem connection, dial-up servers. Modern protocols for multimedia communication: Common Gateway Interface (CGI), multimedia messaging, protocols for multimedia communication – hypertext.

HTML programming language: HTML tags and concepts such as tables, frames, forms and cascading style sheets; hypertext design, XML. Web

services and servers, examples and design of web pages, search engines and indexing. Elements of programming language: JavaScript, dynamic HTML pages. Development and the future communication using Internet. New

technologies.



CIT 322

Introduction to Internet Programming

Course Developer Dr. Ikhu Omoregbe, Nicholas

Course Adapter Afolunso, A. A.
National Open University of Nigeria
Lagos.

Course Editor

Course Co-ordinator Afolunso, A. A.
National Open University of Nigeria
Lagos.



NATIONAL OPEN UNIVERSITY OF NIGERIA

National Open University of Nigeria
Headquarters
14/16 Ahmadu Bello Way
Victoria Island
Lagos

Abuja Annex
245 Samuel Adesujo Ademulegun Street
Central Business District
Opposite Arewa Suites
Abuja

e-mail: centralinfo@nou.edu.ng

URL: www.nou.edu.ng

National Open University of Nigeria 2008

First Printed 2008

ISBN

All Rights Reserved

Printed by

For

National Open University of Nigeria

TABLE OF CONTENTS	PAGE
Introduction.....	1 - 2
What you will learn in this Course.....	2
Course Aims.....	2
Course Objectives.....	2 – 3
Working through this Course.....	3
Course Materials.....	3
Study Units	3 - 4
Textbooks and References	4 - 5
Assignment File.....	5
Presentation Schedule.....	5
Assessment.....	6
Tutor Marked Assignments (TMAs)	6
Examination and Grading.....	6 - 7
Course Marking Scheme.....	7
<i>Course Overview</i>	7 - 8
How to Get the Best from This Course	8
- 9	
Tutors and Tutorials	10
Summary	10 -
11Figure 1	

Introduction

CIT 322 – Introduction to Internet Programming is a three [3] credit unit course of fourteen units. It deals with concepts, architecture, features, and services of the Internet and a comprehensive introduction to current programming models and tools for generating rich Web applications. The Internet makes information instantly and conveniently available worldwide. In this course we made effort to provide the skills required for student to design and deploy solutions on the Internet. A number of tools for constructing Web applications such as Hypertext Markup Language (HTML), Cascading Style Sheets, JavaScript, and Extensible Markup Language have got a very wide coverage in this course. Web services, Cloud Computing, mashups, microformat, folksonomies, and Web 2.0 technologies which are hot topics today in Web application developments have also been taken up. The course is divided into four modules.

Module 1

This module lays the ground for the rest of the course material. It provides fundamental information on the meaning, origin, architecture and services offered by the internet. It also explores network models and the protocols that work behind the scene to display Web pages.

Module 2

Hypertext Markup Language is one of the most important languages used to construct Web pages. This module concentrates on the syntax and the various HTML tags, elements and attributes used in developing Web pages. Some of these include anchor, table, form, image, video, audio etc. The module contains several program segments to demonstrate how HTML codes are written. These are included for the students to learn and acquire the requisite skills to develop other real world applications.

Module 3

Hypertext Markup Language (HTML) was not design to contain tags for formatting a document or making web pages interactive or responsive to users. In this module we explored tools such as Cascading Style Sheets and JavaScript for formatting text and adding interactivity respectively to Web pages. The module contains some sample codes in JavaScript and Cascading Style Sheets that demonstrate their features and capabilities in making Web pages come alive.

Module 4

This module contains information on Search engines and the technologies and tools for developing faster and interoperable Web applications. Hot topics such as Extensible Markup Language (XML), Web Services, Mashups, Application Programming Interface (API), Cloud Computing, Blogs, Podcasting, RSS, ontologies and Web 2.0 technologies were covered also in this Module.

The aim of this course is to equip you with the basic skills of studying and understanding a Internet programming well as lay the foundation of the basic knowledge and tools you need to become a proficient Web content developer /administrator. By the end of the course, you should be able to confidently work in any Internet services providing organization

This Course Guide gives you a brief overview of the course content, course duration, and course materials.

What you will learn in this course

The main purpose of this course is to provide a comprehensive Introduction the fundamentals of Internet technologies and the tools and skills required to build and maintain Web applications. A wide variety of technologies and tools that are used for the construction of Website have been extensively discussed in this course. We intend to achieve this through the following:

Course Aims

- i. Introduce the concepts, features and services of the Internet;
- ii. Explore the architecture for Web applications and the tools for building standard websites;
- iii. Explore and used various HTML tags for web development
- iv. Expose the techniques for writing Cascading Style Sheet as a standardized way of imposing style on the content of HTML tags.
- v. Expose how to write JavaScript which are used for adding interactivity to static pages and how to write XML to make Web applications more interoperable for data storage and exchange
- iv. Expose the current trends in Web application development such as Web services, cloud computing, mashup, RSS, Wiki, etc

Course Objectives

Certain objectives have been set out to ensure that the course achieves its aims. Apart from the course objectives, every unit of this course has set objectives. In the course of the study, you will need to confirm, at the end of each unit, if you have met the objectives set at the beginning of each unit. By the end of this course you should be able:

- to define and discuss the evolution of the Internet
- to explain the meaning of intranet and extranet
- to list the devices use to access the Internet
- to explain the various means of accessing the Internet

- to describe the services offered by the Internet
- to describe the protocols used by the Internet
- to explain the meaning of a computer network
- to discuss the client-server model
- to describe the Web application architecture
- to explain the meaning of Common Gateway Interface
- to discuss the meaning of HTML
- to understand simple HTML codes
- to use Web browsers to display HTML codes
- to write HTML codes using popular tags
- to run HTML codes
- to write HTML codes that enhance the appearance of web pages
- to identify multimedia elements
- to describe different multimedia formats
- to embed multimedia files in HTML codes
- to create link to other web pages
- to design tables organize information using a table
- to create tables using table tags
- to format table by using appropriate table attributes
- to explain the meaning of Forms
- to write HTML codes to process form information
- to explain how to use Form action and Methods
- to discuss and use various form elements
- to explain the meaning and the importance of CSS
- to use CSS format web pages
- to add CSS to HTML files
- to explain the meaning of JavaScript
- to write simple JavaScript programs
- to use arithmetic operators
- to use JavaScript to make web pages interactive
- to implement logical construct with JavaScript
- to implement decision statements with JavaScript
- to explain the meaning of XML
- to explain how to create and modify XML document
- to explain how to process and validate XML document
- to explain how to view XML documents a Web browser
- implements loops with JavaScript
- to explain the meaning of Search Tools
- to describe the components of a Search Engine
- to explain how search engines works
- to discuss how to popularize an a site
- to describe the tools used for Web applications
- to explain the meaning of Web 2.0 Technologies

- to classify and use Web development tools

Working Through This Course

In order to have a thorough understanding of the course units, you will need to read and understand the contents, practise the what you have learnt by studying the network of your organization or proposing one if there is none in existence and be committed to learning and implementing your knowledge.

This course is designed to cover approximately sixteen weeks, and it will require your devoted attention. You should do the exercises in the Tutor-Marked Assignments and submit to your tutors.

Course Materials

These include:

1. Course Guide
2. Study Units
3. Recommended Texts
4. A file for your assignments and for records to monitor your progress.

Study Units

There are fourteen study units in this course:

Module1

- Unit 1: Internet Architecture and Organization
- Unit 2: Internet Services and Communication and Protocol
- Unit 3: Network Model and Web Application Development

Module 2

- Unit 1: Understanding HTML
- Unit 2: HTML Elements
- Unit 3: Tables
- Unit 4: Input Tags and Form Processing

Module 3

- Unit 1: Cascading Style Sheet
- Unit 2: Fundamentals of JavaScript
- Unit 3: Decision and Interactive Statements
- Unit 4: Events and Event Handlers

Module 4

Unit 1: Overview of XML

Unit 2: Unit 1: Search Engines & Tools

Unit 3: The Future Web, Technologies and Development Tools

Make use of the course materials, do the exercises to enhance your learning.

Textbooks and References

1. **Alex Leon and Mathew Leon (1999), Fundamentals of Information Technology, Vikas Publishing House PVT LTD, New Delhi**
2. **Andy Sloane (1999), Computer Communications, Principles and Business Applications, Second Edition, McGraw-Hill, England**
3. **Barbara K. Kaye, Norman J. Medoff (2001) The World Wide Web: A Mass Communication Perspective, Mayfield Publishing Company, USA**
4. **Barrie Sosinsky, Valda Hilley, (2004) Programming the Web: An Introduction, McGraw-Hill, USA**
5. **Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA**
6. **Brian K. Williams, Stacey C. Sawyer, Sarah E. Hustchinson(1999) Using Information Technology: A Practical Introduction to Computers and Communication, Irwin/McGraw-Hill**
7. **Deitel, P. J. and Deitel H.M (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA**
8. **June Cohen (2003), The Unusually Useful Web Book, New Rider, USA**
9. **MacBride K. (2006), Brilliant Internet for the Over 50s, What you need to know about it Pearson Education Limited, England**
10. **Marc D. Miller, Thomas C. Padgett (2003) Web Design Using Macromedia Dreamweaver, McGraw-Hill/ Irwin, USA**
11. **Miller David (2006), Data Communications and Network, McGraw Hill, New York**
12. **Nagpal, D. P (2006) Web Design Technology, Theor y and Technique on the Cutting Edge S. Chand and Company Ltd, New Delhi, India**

13. **Nolan Hester (2005) Creating a Web Page in Dreamweaver, Peachpit Press, Berkeley, USA**
14. **Peter Weverka (2001) Instant Web Pages, Sybex Inc, USA**
15. **Raymond Greenlaw and Ellen Hepp (2001), In-Line/On-Line Fundamentals of the Internet and World Wide Web, McGraw-Hill Learning Center, USA**
16. **Richard Alan McMahon Sr. (2003) Introduction to Networking, McGraw-Hill/Osborne, N.Y, USA**
17. **Robert W. Sebesta (2009), Programming the World Wide Web, Pearson Addison-Wesley, New Jersey, USA**
18. **Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA**
19. Katarzyna J. Macura, Internet and You: Connecting to the Internet
<http://www.rsna.org/Technology/internet2-1.cfm>
20. <http://www.wisegeek.com/what-is-a-web-browser.htm>
21. <http://www.daaq.net/old/javascript/index.php?page=js+iteratives&parent=js+statements>
22. http://books.mcgraw-hill.com/downloads/products/0072194154/0072194154_ch01.pdf
23. http://www.rebol.com/docs/core23/rebolcore-13.html#_Toc487519913
24. <http://www.tizag.com/cssT/selector.php>
25. http://www.w3schools.com/css/css_syntax.asp

Assignments File

These are of two types: the self-assessment exercises and the Tutor-Marked Assignments. The self-assessment exercises will enable you monitor your performance by yourself, while the Tutor-Marked Assignment is a supervised assignment. The assignments take a certain percentage of your total score in this course. The Tutor-Marked Assignments will be assessed by your tutor within a specified period. The examination at the end of this course will aim at determining the level of mastery of the subject matter. This course includes seventeen Tutor-Marked Assignments and each must be done and submitted accordingly. Your best scores however, will be recorded for you. Be sure to send these assignments to your tutor before the deadline to avoid loss of marks.

Presentation Schedule

The Presentation Schedule included in your course materials gives you the important dates for the completion of tutor marked assignments and attending tutorials. Remember, you are required to submit all your assignments by the due date. You should guard against lagging behind in your work.

Assessment

There are two aspects to the assessment of the course. First are the tutor marked assignments; second, is a written examination.

In tackling the assignments, you are expected to apply information and knowledge acquired during this course. The assignments must be submitted to your tutor for formal assessment in accordance with the deadlines stated in the Assignment File. The work you submit to your tutor for assessment will count for 30% of your total course mark.

At the end of the course, you will need to sit for a final three-hour examination. This will also count for 70% of your total course mark.

Tutor Marked Assignments (TMAS)

There are fourteen tutor marked assignments in this course. You need to submit all the assignments. The total marks for the best four (4) assignments will be 30% of your total course mark.

Assignment questions for the units in this course are contained in the Assignment File. You should be able to complete your assignments from the information and materials contained in your set textbooks, reading and study units. However, you may wish to use other references to broaden your viewpoint and provide a deeper understanding of the subject.

When you have completed each assignment, send it together with form to your tutor. Make sure that each assignment reaches your tutor on or before the deadline given. If, however, you cannot complete your work on time, contact your tutor before the assignment is done to discuss the possibility of an extension.

Examination and Grading

The final examination for the course will carry 70% of the total marks available for this course. The examination will cover every aspect of the course, so you are advised to revise all your corrected assignments before the examination.

This course endows you with the status of a teacher and that of a learner. This means that you teach yourself and that you learn, as your learning capabilities would allow. It also means that you are in a better position to determine and to ascertain the what, the how, and the when of your course learning. No teacher imposes any method of learning on you.

The course units are similarly designed with the introduction following the table of contents, then a set of objectives and then the concepts and so on.

The objectives guide you as you go through the units to ascertain your knowledge of the required terms and expressions.

Course Marking Scheme

This table shows how the actual course marking is broken down.

Assessment Marks	
Assignment 1- 4	Four assignments, best three marks of the four count at 30% of course marks
Final Examination	70% of overall course marks
Total	100% of course marks

Table 1: Course Marking Scheme

Course Overview

Unit	Title of Work	Weeks	Assessment
		Activity	(End of Unit)
	Course Guide	Week 1	
	Module 1		
1	Internet Architecture and Organization	Week 1	Assignment 1
2	Internet Services and Communication and Protocol	Week 2	Assignment 2
3	Network Model and Web Application	Week 3	Assignment 3

	Development		
	Module 2		
1	Understanding HTML	Week 4	Assignment 4
2	HTML Elements	Week 5	Assignment 5
3	Tables	Week 6	Assignment 6
4	Input Tags and Form Processing	Week 7	Assignment 7
	Module 3		
1	Cascading Style Sheet	Week 8 & 9	Assignment 8
2	Fundamentals of JavaScript	Week 10	Assignment 9
3	Decision and Interactive Statements	Week 11	Assignment 10
4	Events and Event Handler in JavaScript	Week 12	Assignment 11
	Module 4		
1	Understanding XML	Week 13	Assignment 12
2	Search Engines & Tools	Week 14	Assignment 12
3	The Future Web, Technologies and Development Tools	Week 15	Assignment 13
	Revision	Week 16	
	Examination	Week 17	
Total		17 weeks	

How to get the best from this course

In distance learning the study units replace the university lecturer. This is one of the great advantages of distance learning; you can read and work through specially designed study materials at your own pace, and at a time and place that suit you best. Think of it as reading the lecture instead of listening to a lecturer. In the same way that a lecturer might set you some reading to do, the study units tell you when to read your set books or other material. Just as a lecturer might give you an in-class exercise, your study units provide exercises for you to do at appropriate points.

Each of the study units follows a common format. The first item is an introduction to the subject matter of the unit and how a particular unit is integrated with the other units and the course as a whole. Next is a set of learning objectives. These objectives enable you know what you should be able to do by the time you have completed the unit. You should use these objectives to guide your study. When you have finished the units you must go back and check whether you have achieved the objectives. If you make a habit of doing this you will significantly improve your chances of passing the course.

Remember that your tutor's job is to assist you. When you need help, don't hesitate to call and ask your tutor to provide it.

1. Read this Course Guide thoroughly.
2. Organize a study schedule. Refer to the 'Course Overview' for more details. Note the time you are expected to spend on each unit and how the assignments relate to the units. Whatever method you chose to use, you should decide on it and write in your own dates for working on each unit.
3. Once you have created your own study schedule, do everything you can to stick to it. The major reason that students fail is that they lag behind in their course work.
4. Turn to Unit 1 and read the introduction and the objectives for the unit.
5. Assemble the study materials. Information about what you need for a unit is given in the 'Overview' at the beginning of each unit. You will almost always need both the study unit you are working on and one of your set of books on your desk at the same time.
6. Work through the unit. The content of the unit itself has been arranged to provide a sequence for you to follow. As you work through the unit you will be instructed to read sections from your set books or other articles. Use the unit to guide your reading.
7. Review the objectives for each study unit to confirm that you have achieved them. If you feel unsure about any of the objectives, review the study material or consult your tutor.
8. When you are confident that you have achieved a unit's objectives, you can then start on the next unit. Proceed unit by unit through the course and try to pace your study so that you keep yourself on schedule.
9. When you have submitted an assignment to your tutor for marking, do not wait for its return before starting on the next unit. Keep to your schedule. When the assignment is returned, pay particular attention to your tutor's comments, both on the tutor-marked assignment form and also written on the assignment. Consult your tutor as soon as possible if you have any questions or problems.
10. After completing the last unit, review the course and prepare yourself for the final examination. Check that you have achieved the unit objectives (listed at the beginning of each unit) and the course objectives (listed in this Course Guide).

Tutors and Tutorials

There are 12 hours of tutorials provided in support of this course. You will be notified of the dates, times and location of these tutorials, together with the name and phone number of your tutor, as soon as you are allocated a tutorial group.

Your tutor will mark and comment on your assignments, keep a close watch on your progress and on any difficulties you might encounter and provide assistance to you during the course. You must mail or submit your tutor-marked assignments to your tutor well before the due date (at least two working days are required). They will be marked by your tutor and returned to you as soon as possible.

Do not hesitate to contact your tutor by telephone, or e-mail if you need help. The following might be circumstances in which you would find help necessary. Contact your tutor if:

- you do not understand any part of the study units or the assigned readings,
- you have difficulty with the self-tests or exercises,
- you have a question or problem with an assignment, with your tutor's comments on an assignment or with the grading of an assignment.

You should try your best to attend the tutorials. This is the only chance to have face to face contact with your tutor and to ask questions which are answered instantly. You can raise any problem encountered in the course of your study. To gain the maximum benefit from course tutorials, prepare a question list before attending them. You will learn a lot from participating in discussions actively.

Summary

Introduction to Internet Programming introduces you to basic principles, concepts and features of Internet technologies in addition to the skills for developing Web applications. The skills you need to understand the basics of Internet programming are intended to be acquired in this course. The content of the course material was planned and written to ensure that you acquire the proper knowledge and skills for the appropriate situations. Some real life problems were mentioned or solved in order for you to create some of your own. The essence is to get you to acquire the necessary knowledge and competence. We hope to have achieved that.

I wish you success with the course and hope that you will find it both interesting and useful.

UNIT 1: INTERNET ARCHITECTURE AND ORGANIZATION

CONTENTS

- 1.0 INTRODUCTION
- 2.0 OBJECTIVES
- 3.0 MAIN CONTENT
 - 3.1 The Definition and Evolution of the Internet
 - 3.2 Internet Access
 - 3.2.1 Dial-up Connection
 - 3.2.2 ISDN
 - 3.2.3 Cable TV Connection
 - 3.2.4 Digital Satellite Connection
- 4.0 CONCLUSION
- 5.0 SUMMARY
- 6.0 TUTOR MARKED ASSIGNMENT
- 7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to define and discuss the evolution of the Internet
- to explain the meaning of intranet and extranet
- to list the devices used to access the Internet
- to explain the various means of accessing the Internet

3.0 MAIN CONTEXT

3.1 The Definition and Evolution of the Internet

The Internet also referred to as the net, in simplest terms, consists of a large group of millions of computers around the world that are connected to one another. It is a network of networks that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical technologies such as phone lines, fibre optic lines, coaxial cable, satellites, and wireless connections. The Internet seems to be everywhere today with many people and devices connected to it. When connected to the Internet people can access services such as online shopping, listen to radio and TV broadcast, chat, send mail, access information, read newspaper and etc. Today Internet is not only accessed from regular stationary computer but also from mobile / portable devices such as Personal digital assistants (PDAs), cell phones, netbook, iPod, iPad, Palm Pilots etc.



Figure 1.1: PDA access to the Internet

The Internet originated as a proposal from the Advanced Research Project Agency (ARPA). The idea was to see how computers connected in a network i.e (ARPANET) could be used to access information from research facilities and universities. In 1969, four computers (located at UCLA, Stanford Research Institute, University of California Santa Barbara and the University of Utah) were successfully connected. As time went on, other networks were connected. With four nodes by the end of 1969, the ARPANET spanned the continental U.S. by 1971 and had connections to Europe by 1973. Though the Interconnected Network, or Internet, was originally limited to the military, government, research, and educational purposes it was eventually opened to the general public. And today there are hundreds of millions of computers and other devices connected to the Internet worldwide.

Other definitions that are closely related to the term Internet are intranet and extranet.

Intranet

The term intranet is used to describe a networked of PC without any PC on the network connected to the world outside of the intranet. The Intranet resides behind a firewall and, if it allows access from the Internet, will be an Extranet. The firewall helps to control access between the intranet and Internet so that only authorized users will have access to the intranet. These are usually people who are members of the same company or organisation. Like the Internet itself, intranets are used to share information. Secure intranets are now the fastest-growing segment of the Internet because they are much less expensive to build and manage than private network or proprietary protocols.

Extranet

Extranets are becoming a very popular means for business partners to exchange information. An Extranet is a term used to refer to an intranet that is partially accessible to authorised outsiders. Privacy and security are important issues in extranet use. A firewall is usually provided to help control access between the Intranet and Internet. In this case, the actual server will reside behind a firewall. The level of access can be set to different levels for individuals or groups of outside users.

3.2 INTERNET ACCESS

To connect to the vast resources on the Internet, you connect your computer to a computer that is on the Internet usually one run by an ISP. You can connect your computer by using a dial-up phone line, which is how most home users connected to the Internet during the 1990s. If you need to connect at higher speeds than a regular phone line allows, you can get a high-speed, assuming that your phone company offers them. You may have three options, depending on what your phone company offers: DSL, ISDN, or a leased line (such as a T1). Connection Speeds. the Internet by using a cable TV connection. Rural users may consider installing a satellite dish for Internet connections, whereas urban users may have access to a wireless connection. In offices, most computers connect via a local area network (LAN), and many homes are beginning to connect their computers into Internet-connected LANs, too. High-speed Internet connections, including DSL, ISDN, leased lines, cable Internet, and satellite, are all called broadband connections.

There are four ways of connecting a client computer to the vast resources on the Internet: a dial-up connection using a telephone line or an Integrated Services Digital Network (ISDN), a Digital Subscriber Line (DSL), a cable TV connection and a satellite connection.

3.2.1 Dial-up Connection

Dial-up Internet access is a form of Internet access that uses the facilities of the public switched telephone network (PSTN) to establish a dialed connection to an Internet service provider (ISP) via telephone lines. The user's computer or router uses an attached modem to encode and decode Internet Protocol packets and control information into and from analog audio frequency signals, respectively. The term was coined during the early days of computer telecommunications when modems were needed to connect terminals or computers running terminal emulator software to mainframes, minicomputers, online services and bulletin board systems via a telephone line. To use a dial-up account, you need a modem. A modem (modulator-demodulator) is a device that modulates an analog carrier signal to encode digital information, and also demodulates such a carrier signal to decode the transmitted information. To distinguish dial-up modems from newer, high-speed modems, they are could also be called analog modems or dial-up modems. Most computers come with an internal modem and most ISPs support modems at speeds of 28.8 Kbps and 56 Kbps. With dial-up, you connect only when you want to use Internet services and disconnect (hang up) when you are done. This type of data transmission is similar to using the telephone to make a call. The client computer modem dials the preprogrammed phone number for a user's Internet Service Provider (ISP) and connects to one of the ISP's modems. Once the ISP has verified the user's account, a connection is established and data can be transmitted. The communication ends when either modem hangs up. Dial up connections is not expensive (it costs no more than a local telephone call) but the speed is usually low at about 28kps – 46kps because of the limitations of analog phone lines and telephone company switches.

3.2.2 ISDN

Integrated Services Digital Network (ISDN) is a set of communications standards for simultaneous digital transmission of voice, video, data, and other network services over the

traditional circuits of the public switched telephone network. It allows dial up into the Internet at speeds ranging from 64 to 128 kbps. For this connection to be available, telephone companies would have to install special ISDN digital switching equipment. The ISDN service intended for residential use is Basic Rate Interface (BRI). On one ISDN line, BRI provides two 64-Kbps channels, or B(bearer) channels, and one 16-Kbps channel, or D(data) channel. The D channel is mostly used for signalling such as to indicate that the line is busy. The B channels are where the action is. Two B channels can be combined to have a 128-Kbps line to the Internet. This is roughly twice the speed of the fastest analog modem, 56 Kbps. To connect to your ISP via ISDN you need to confirm the availability of the access and this will require you to have an ISDN adapter. ISDN lines are more expensive than normal phone lines, so the telephone rates are usually higher.

3.2.3 Cable TV Connection

This is a connection made to the Internet via Cable a TV modem. The modem is designed to operate over cable TV lines. Because the coaxial cable used by cable TV provides much greater bandwidth than telephone lines, a cable modem can be used to achieve extremely fast speed as high as 128 kbps to 10 mbps to the World Wide Web. This combined with the fact that millions of homes are already wired for cable TV in developed countries have made the cable modem something of a holy grail for Internet and cable TV companies. The services offered are usually at low cost for unlimited, always connected" access. However, there are a number of technical difficulties in this type of connection. The problem is that the cable network was designed to move information in one direction, from the broadcaster to the user. Downstream speeds have been very impressive such that the line can theoretically bring you data as fast as 30 Mbps but upstream speed depends on line quality. The Internet, however, is a two-way system where data also needs to flow from the client to the server. In addition, it is still unknown whether the cable TV networks can handle the traffic that would ensue if millions of users began using the system for Internet access. Large cable companies are spending money to upgrade their networks to hybrid fiber-coaxial (HFC) to better handle two-way traffic. Smaller providers can't afford the upgrade, so they have to use a phone line at 28.8 Kbps for upstream data. Another issue bothers on security and the need to either share or not share files amongst users.

DSL (Digital Subscriber Line)

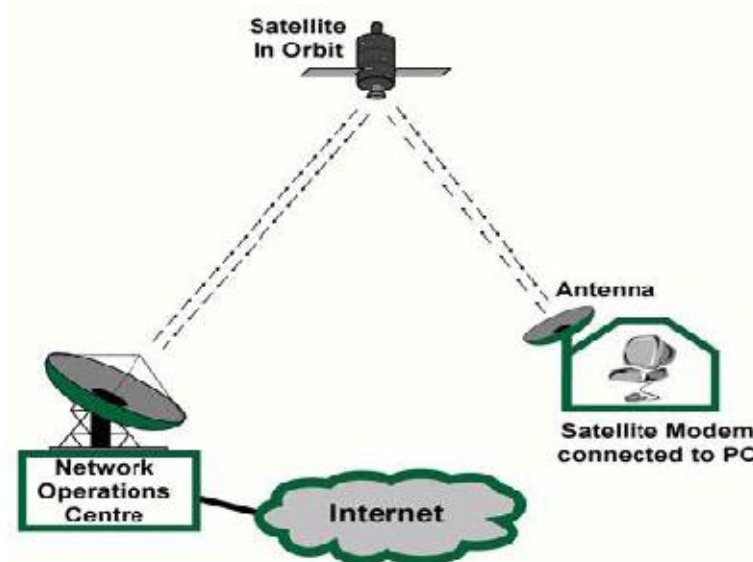
Digital Subscriber Line (DSL) is a family of technologies that provides digital data transmission over the wires of a local telephone network. DSL service is delivered simultaneously with regular telephone on the same telephone line. DSL uses a different part of the frequency spectrum than analog voice signals, so it can work in conjunction with a standard analog telephone service, providing separate voice and data "channels" on the same line. SDSL (Symmetric DSL) is the type of DSL that offers the same bandwidth capability in both directions while ADSL (Asymmetric DSL) is the type of DSL that provides different bandwidths in the upstream and downstream directions. Most DSL lines are actually ADSL (Asymmetric Digital Subscriber Line). ADSL is optimized for the way many people use the Internet: more downloads than uploads. The line is asymmetric, because it has more capacity for data received by your computer (such as graphics, video, audio, and software upgrades) than for data that you send (such as e-mail and browser commands). The data throughput of consumer DSL services typically ranges from 256 kbit/s to 40 Mbit/s in the direction to the customer (downstream),

depending on DSL technology, line conditions, and service-level implementation. In ADSL, the data throughput in the upstream direction, (i.e. in the direction to the service provider) is lower, hence the designation of asymmetric service. In Symmetric Digital Subscriber Line (SDSL) service, the downstream and upstream data rates are equal

Unlike cable modem technology, DSL provides a point-to-point connection to ISP. Somehow, this technology seems to be both more secure and less prone to local traffic fluctuations than its cable rival.

3.2.4 Digital Satellite Connection

Digital Satellite Systems (DSS), or direct broadcast satellite, lets you get Internet information by satellite. Satellite Internet systems are an excellent, although rather pricey, option for people in rural areas where Digital Subscriber Line (DSL) and cable modem connections are not available. A satellite installation can be used even where the most basic utilities are lacking, if there is a generator or battery power supply that can produce enough electricity to run a desktop computer system. The two-way satellite Internet option offers an always-on connection that bypasses the dial-up process. In a two-way satellite Internet connection, the upstream data is usually sent at a slower speed than the downstream data arrives. Thus, the connection is asymmetric. A dish antenna, measuring about two feet high by three feet wide by three feet deep, transmits and receives signals. Uplink speeds are nominally 50 to 150 Kbps for a subscriber using a simple computer. The downlink occurs at speeds ranging from about 150 Kbps to more than 1200 Kbps, depending on factors such as Internet traffic, the capacity of the server, and the sizes. The main advantage of the Satellite technology over cable modems and DSL is accessibility. Satellite connections are faster than dial up and ISDN. But not as fast as cable modems or DSL services, which both can provide more than megabits of bandwidth. Also, cable and DSL access methods are cheaper.



Equipment required for satellite connection includes installation of a mini-dish satellite receiver and a satellite modem. Satellite systems are also prone to rain fade (degradation during heavy precipitation) and occasional brief periods of solar interference.

4.0 SUMMARY

The general rule about the Internet connection is the faster, the better. The bandwidth and transfer rate determine how quickly pictures, sounds, animation and video clips will be downloaded. Since multimedia and interactivity make the Internet such an exciting tool for information sharing, the speed is the key. Dial-up access provides an easy and inexpensive way for users to connect to the Internet, however, it is a slow-speed technology and most users are no longer satisfied with dial-up or ISDN connections. Fortunately, the broadband access, we once dreamed of, is now possible with TV cable, DSL and satellite links.

5.0 CONCLUSION

The Internet has remained a dominant means of communication over the past decade and represents one of the most remarkable developments in the technological history of the world. It began as a medium for exchanging files by academia and has become a nearly ubiquitous phenomenon that has transformed almost every aspect of daily life. The Internet has made information available in a quick and easy manner, publicly accessible and within easy reach via the connections infrastructure discussed in this unit. In the next unit we shall look at some of the services available on the internet and the enabling protocols.

6.0 TUTOR MARKED ASSIGNMENTS

- a) Briefly explain the origin of the Internet
- b) List three (3) examples of mobile devices that can be used to access the Internet
- c) Discuss the four (4) major ways of connecting a client computer to the vast resources on the Internet?

7.0 REFERENCES /FURTHER READING

Alex Leon and Mathew Leon (1999), Fundamentals of Information Technology, Vikas Publishing House PVT LTD, New Delhi

Andy Sloane (1999), Computer Communications, Principles and Business Applications, Second Edition, McGraw-Hill, England

Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Brian K. Williams, Stacey C. Sawyer, Sarah E. Hustchinson(1999) Using Information Technology: A Practical Introduction to Computers and Communication, Irwin/McGraw-Hill

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

Katarzyna J. Macura, Internet and You: Connecting to the Internet

<http://www.rsna.org/Technology/internet2-1.cfm>

http://books.mcgraw-hill.com/downloads/products/0072194154/0072194154_ch01.pdf

<http://www.wisegeek.com/what-is-a-web-browser.htm>

UNIT 2: INTERNET SERVICES AND COMMUNICATION PROTOCOLS

CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES

3.0 MAIN CONTENT

3.1 Services on the Internet

3.1.1 World Wide Web

3.1.2 Electronic Mail

3.2 Communication Protocols

3.2.1 Transmission Control Protocol

3.2.2 Internet Protocol

3.3.3 User Datagram Protocol

3.3.4 Hypertext Transfer Protocol (HTTP)

3.3.5 Email Protocols

3.3.6 File Transfer Protocol (FTP)

|3.3.7 Real Time Streaming Protocol (RTSP)

4.0 CONCLUSION

5.0 SUMMARY

6.0 TUTOR MARKED ASSIGNMENT

7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

The Internet is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide. The Internet offers access to data graphics, sound, software, text, to people through a variety of services and tools for communications and data exchange.

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to describe the services offered by the Internet
- to describe the protocols used by the Internet

3.0 MAIN CONTENT

3.1 Services on the Internet

The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail

3.1.1 World Wide Web

The World Wide Web is a repository of information spread all over the world and together for easy access. It is made up of documents called pages that combine text, pictures, forms, sound, animation and hypertext links into rich communication medium. For several user it is the most exciting aspect of the internet and has accelerated the growth of the Internet by giving it an easy to use, point and click, graphical interface. Users are attracted to the WWW because of its interactive nature. The WWW project was initiated by CERN (European Laboratory for Particle Physics) to create a system to handle distributed resources necessary for scientific research as grown today to become many things to millions of users. It is used as a business place, art gallery, social medium, broadcast medium, library, community centre, school, religious centre, advertise house, publishing house etc.

Most Web pages are prepared using the Markup languages such as Hypertext Markup language (HTML). The document's author can specifically code sections of the documents to point to another information resources. These specially coded sections are referred to as hypertext links. Users viewing a web page can select the hyperlink and retrieve or connect to the information resources that the link points to. Hypertext link can lead to other documents, sound, images, databases (like library catalogs), email address, etc. Figure 2.1 shows the homepage of the National Open University of Nigeria.



Figure 2.1: NOUN's Homepage

Client using a browser such Internet explorer, Firefox or Google chrome can access a service using a server. However, the service provided is distributed over many locations called website.

3.1.2 Electronic Mail

Electronic mail, commonly called email or e-mail, is a method of exchanging digital messages from an author to one or more recipients. Email operates across the Internet or other computer network. An email message consists of three components, the message envelope, the message header, and the message body. The message header contains control information, including,

minimally, an originator's email address and one or more recipient addresses. Usually descriptive information is also added, such as a subject header field and a message submission date/time stamp. To be able to use email you should have an email address which is created at an Internet Service Provider or on a Website such as yahoo, Google, and hotmail. Most e-mail addresses are set up like this: it is your username, then an @ ('at') symbol, and then a domain name (something .com, .net, or .org in most cases). When you send e-mail to others, Simple Mail Transfer Protocol (SMTP) is used. When you receive e-mail, Post Office Protocol (POP, currently POP3) and Internet Message Access Protocol (IMAP) can be used.

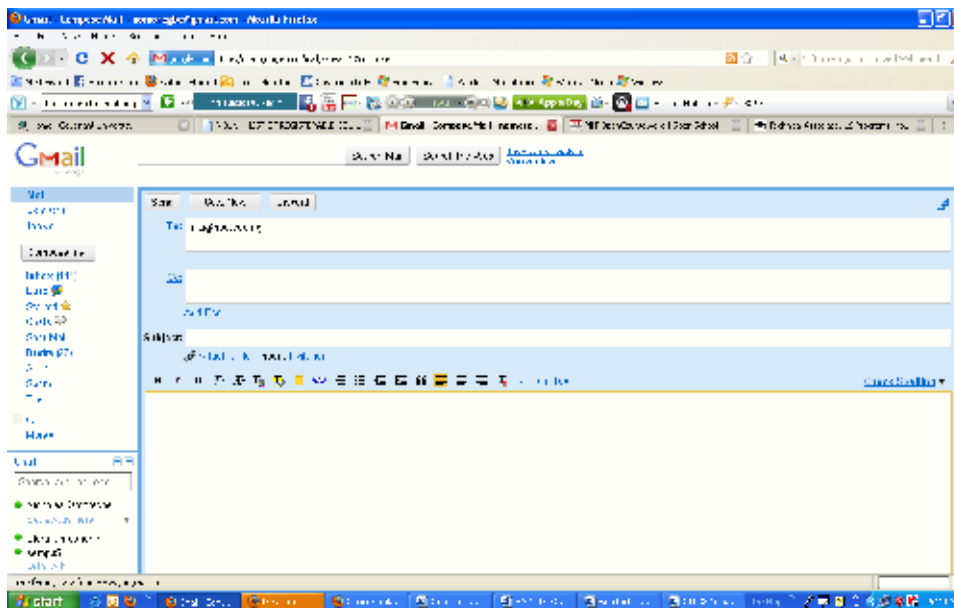


Figure 2.2: Gmail

3.2 COMMUNICATIONS PROTOCOLS

Protocols are rules that describe how client and servers communicate with each other over a network. There is no single protocol that makes the Internet and Web work. A number of protocols with unique functions are required. We will examine some of them

The most commonly used protocols are:

- Transmission Control/Internet Protocol (TCP/IP)
- File Transfer Protocol (FTP)
- Hypertext Transfer Protocol (HTTP)
- Email Protocol

3.2.1 Transmission Control Protocol

The Transmission Control Protocol (TCP), is one of the core protocols of the Internet Protocol Suite. It provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. TCP is the protocol that major Internet applications

such as the World Wide Web, email, remote administration and file transfer rely on. Applications, which do not require reliable data stream service, may use the User Datagram Protocol (UDP), which provides a datagram service that emphasizes reduced latency over reliability. TCP is optimized for accurate delivery rather than timely delivery, and therefore, TCP sometimes incurs relatively long delays (in the order of seconds) while waiting for out-of-order messages or retransmissions of lost messages. It is not particularly suitable for real-time applications such as Voice over IP. For such applications, protocols like the Real-time Transport Protocol (RTP) running over the User Datagram Protocol (UDP) are usually recommended instead.

3.2.2 Internet Protocol

The Internet Protocol (IP) is a set of rules that are more concerned with sending a message to the correct address than with whether the data actually makes it to that receiver. It is therefore, a connectionless protocol, which means that it is an unreliable protocol. IP works by exchanging pieces of information called packets. A packet is a sequence of octets and consists of a header followed by a body. The header describes the packet's destination and, optionally, the routers to use for forwarding until it arrives at its destination. The body contains the data IP is transmitting. IP will send it information regardless of whether the receiver is there or not. Of primary importance to the IP's set of rules is the creation and maintenance of an addressing scheme, known as IP addressing. IP is responsible for selecting the best route for each message to travel on its way to the receiver. TCP is required to complement IP for effective delivery of information over the Internet.

Although it did not start out that way, all computers now connecting to the Internet must do so using an IP address. Each IP address is a unique 4-byte (or 32-bit) number formatted such that each byte (or 8-bit segment) is separated by a period. An example of would be 100.100.100.100

Each device connected to the internet has a unique numeric IP address. These addresses consist of a set of four group of numbers, called octet. The current version IP, IPv4 uses 32 bits while IPv6 uses 128 bits. The format of IPv4 is xxx.xxx.xxx.xxx where xxx is a value from 0 to 255. The IP address may correspond to a domain name. The domain name system (DNS) associate these IP address with text-based URLs and domain names you type into a Web browser address box. It may be easier to type the URL than the IP address. IPv6 is the latest version of the IP routing protocol. It became necessary to introduce a new protocol in order to accommodate the greater demands being placed on the Internet by increasing user and device access. The major features of this version are: changes in the new version are:

- More addresses—this is done by increasing the IP address size from 32 to 128 bits. Thus, this version can accommodate everyone PC, cell phone, PDA, Automobile and other devices that may be connected to the internet.

- Simplified IP headers –There is a reduction in the number of header fields needed in IP packets for IPv6 compared to IPv4

- Additional security features –IPV6 provides greater support for privacy and security.

IPv6 is more efficient

Domain Name Service (DNS)

An alternative to using the IP address method for locating resources on the Internet is by using the Domain Name Service (DNS) combined with a site's Uniform Resource Locator (URL). URLs, are especially formatted names like www.nou.edu.ng

DNS is like a giant phone book where you can find an IP address knowing the URL. Or you can provide an IP address and the DNS server will like it to the URL. Thousands of DNS servers exist to furnish users with IP addresses. When a user types a URL into a web browser, a request is sent to any listening DNS server to furnish the corresponding IP address. As long as the URL is listed in a listening DNS sever, the correct IP address will be returned and the communication will occur.

3.3.3 User Datagram Protocol

An alternative to TCP for communication in the Transport layer is User Datagram Protocol (UDP), UDP is a connectionless protocol (like IP) that operates at the transport layer. It can actually be faster than TCP in some instances because, as a connectionless protocol, it does not have to open a connection with the receiver, and it does not have to do any error correction. Both of these functions are performed by TCP- a connection-oriented, or reliable , protocol – and they take additional overhead in the form of added steps, and they may slow down transmission as a result. However, in cases of large message and faulty connections, errors may occur and retransmission may ultimately make TCP faster than UDP in the long run. UDP does no checks to ensure receipts so it never does automatic retransmission. Missed messages may, therefore, result in slower communication over UDP.

3.3.4 Hypertext Transfer Protocol (HTTP)

HTTP, is a set of rules for exchanging files such as text, graphics images, sound, video and other multimedia files on the Web. Web browsers and Web Servers usually use this protocol. HTTP is based on the client/server principle. HTTP allows computer A (the client) to establish a connection with Computer B (the server) and make a request. The server accepts the connection initiated by the client and sends back a response. An http request identifies the resources that the client is interested in and tells the server the server what action to take on the resources. When the user of a Web browser requests a file by typing a Web site address or clicking a hyperlink, the browser builds an HTTP request and sends it to the server. The Web server in the destination machine receives the request, does any necessary processing, and responds with the requested file and any associated media files. To retrieve a Web page, the browser sends a request to a Web server using HTTP. On receiving the request, the server interprets it, sometimes using a CGI script (see CGI - Common Gateway Interface), and sends back data. This data can be just about anything, including HTML, text, images, programs, and sound.

3.3.5 Email Protocols

Two main servers are required for email messages to be sent and delivered successfully. These are –incoming mail server and an outgoing mail server. Incoming e-mail messages are sent to an e-mail server that stores messages in the recipient's email box. The user retrieves the messages with an e-mail client that uses one of a number of e-mail retrieval protocols. Some clients and servers preferentially use vendor-specific, proprietary protocols, but most support the Internet standard protocols, SMTP for sending e-mail and POP and IMAP for retrieving e-mail, allowing interoperability with other servers and clients

SMTP - Simple Mail Transport Protocol

The Simple Mail Transport Protocol (SMTP) controls the transfer of email messages on the Internet. SMTP defines the interaction between Internet hosts that participate in forwarding email from a sender to its destination.

POP - Post Office Protocol

The Post Office Protocol (POP) allows you to fetch email that is waiting in a mail server mailbox. POP defines a number of operations for how to access and store email on your server.

IMAP - Internet Message Access Protocol

It is an Internet protocol that allows an email client to access email on a remote mail server.

3.3.6 File Transfer Protocol (FTP)

FTP is a set of rules that allows files to be exchanged between computers on the Internet. The File Transfer Protocol (FTP) is used widely on the Internet for transferring files to and from a remote host. FTP is commonly used for uploading pages to a Web site and for providing online file archives. Unlike HTTP, which is used by Web browser to request Web pages and associated files in order to display a Web page, FTP is used simply to move files from one computer to another. Web developers commonly use FTP to transfer Web pages files from their computers to Web servers. FTP is also commonly used to download programs and files from other servers to individual computers. Access to FTP servers can be open or closed. Open access allows anyone to login to the site and download files. This is called anonymous access and it is used frequently for public file archives. Closed access requires that you provide a username and password to download and upload files. This is the mode of operation for uploading Web pages to a Web site. FTP uses two well-known TCP ports: port 21 is used for the control connection, while port 20 is used for the data connection.

3.3.7 Real Time Streaming Protocol (RTSP)

The Real Time Streaming Protocol (RTSP) is a network control protocol designed for use in entertainment and communications systems such as webcasting to control streaming media servers. Webcasting is the delivery of multimedia data in streaming format across the Internet. Essentially, webcasting is broadcasting over the Internet. A webcast can be use to deliver live or on-demand educational and training content or facilitate collaborative applications such as streaming, chat within an organization. RTSP is used for establishing and controlling media sessions between end points. Clients of media servers issue VCR-like commands, such as play and pause, to facilitate real-time control of playback of media files from the server. The transmission of streaming data itself is not a task of the RTSP protocol. To stream data from one location to another simply means that when data is accessed from a source or upon initiation of a data transmission from a source, not all of the data is delivered to the recipient before the data can begin to be viewed at the destination. Steaming utilizes underlying transport and control protocol such as RTP, UDP, and RTCP(Real-Time Transport Control Protocol). RTCP provides out-of-band statistics and control information for an RTP flow. It partners RTP in the delivery and packaging of multimedia data, but does not transport any media streams itself. RTSP is much like HTTP, except that where HTTP will deliver a file from a Web server and then release the connection until the next file is requested, RTSP maintains the connection between a streaming server and the client that is receiving the streamed data.

4.0 CONCLUSION

Most traditional communications media including telephone, music, film, and television are fast being reshaped or redefined by the Internet, giving birth to new services such as Voice over Internet Protocol (VoIP) and IPTV. Newspaper, book and other print publishing are adapting to Web site technology, or are reshaped into blogging and web feeds. The Internet has enabled or accelerated new forms of human interactions through instant messaging, Internet forums, and social networking. Online shopping has boomed both for major retail outlets and small artisans and traders. Business-to-business and financial services on the Internet affect supply chains across entire industries

5.0 SUMMARY

The internet provides lots of services which are made possible by communication protocols. Transmission Control Protocol/Internet Protocol (TCP/IP) for an example has been adopted as the official communication protocol of the Internet. TCP/IP is actually a collection of protocols, or rules, that governs the way data travels from one machine to another across network. TCP and IP have different functions that work together to ensure reliable communication over the Internet. Other protocols covered in this unit are Email, File Transfer Protocol (FTP) and the Real Time Streaming Protocol (RTSP) which is designed for use in entertainment and communications systems such as webcasting to control streaming media and the

6.0 TUTOR MARKED ASSIGNMENTS

- a) Explain the meaning of the term protocols
- b) Discuss the following protocols
 - i) TCP/IP
 - ii) RTSP
 - iii) HTTP

7.0 REFERENCES / FURTHER READINGS

Alex Leon and Mathew Leon (1999), Fundamentals of Information Technology, Vikas Publishing House PVT LTD, New Delhi

Andy Sloane (1999), Computer Communications, Principles and Business Applications, Second Edition, McGraw-Hill, England

Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Brian K. Williams, Stacey C. Sawyer, Sarah E. Hustchinson(1999) Using Information Technology: A Practical Introduction to Computers and Communication, Irwin/McGraw-Hill

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

Katarzyna J. Macura, Internet and You: Connecting to the Internet

<http://www.rsna.org/Technology/internet2-1.cfm>

http://books.mcgraw-hill.com/downloads/products/0072194154/0072194154_ch01.pdf

<http://www.wisegeek.com/what-is-a-web-browser.htm>

UNIT 3: NETWORK MODEL AND WEB APPLICATION DEVELOPMENT

CONTENTS

- 1.0 INTRODUCTION
- 2.0 OBJECTIVES
- 3.0 MAIN CONTENT
 - 3.1 Network Overview
 - 3.2 The Client Server Model
 - 3.3 Types of Server
 - 3.4 Web Application
 - 3.4.1 The Web Browser
 - 3.4.2 The Web Server
 - 3.5 OSI Reference Model Concept
 - 3.6 TCP/IP Protocols Suit
 - 3.7 Common Gateway Interface
- 4.0 CONCLUSION
- 5.0 SUMMARY
- 6.0 TUTOR MARKED ASSIGNMENT
- 7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

A CLIENT: A client is a program running on the local machine requesting service from a server. A client program is started by the user or another application program and terminates when the service is complete

A SERVER – is a program running on the remote machine providing service to the clients. When it starts, it opens the door for incoming request from clients, but it never initiates a service until it is requested to do so.

A network of networks, or Internet, refers to a group of two or more networks that are interconnected physically capable of communication and sharing data with each other and able to act together as a single network. Machine on one network can communicate with machines on other networks, and data, file and other information back and forth. For this to work, the systems must follow some set of rules or protocols. This is a language or software that enables different types of machine on separate network to communicate and exchange information. The Internet uses the TCP/IP protocol. The Internet offers access to data graphics, sound, software, text, and people through a variety of services and tools for communications and data exchange. Some services available on the Internet are as follows:

- Remote login (telnet)
- File transfer (ftp)
- Electronic mail (e-mail)

News (USENET or network news)
Hypertext (www)

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to explain the meaning of a computer network
- to discuss the client-server model
- to describe the Web application architecture
- to explain the meaning of Common Gateway Inter face

3.1 Network Overview

A network consists of two or more computers connected for the purpose of communicating and sharing resources. A network consists of a group of two or more computer systems linked together. There are many types of computer networks, including:

Local-area networks (LANs) : This describes computers that are geographically close together (that is, in the same building).

Wide-area networks (WANs) : This describes computers that are farther apart and connected by telephone lines or radio waves.

Campus-area networks (CANs): This describes computers that are within a limited geographic area, such as a campus or military base.

Metropolitan-area networks (MANs): This describes data network designed for a town or city.

Home-area networks (HANs): This describes a network contained within a user's home that

Computers on a network are sometimes called nodes. The common components of a network are:

- Server
- Client workstation computer(s)
- Shared devices such as printers
- Networking devices (hub) and the media that connect them

3.2 The Client Server Model

The term client / server date from the last millennium (1980s) and refers to personnel computers joined by a network. Client/Server can also describe a relationship between two computer programs- the client and the server. The client/server technology evolved as a result of downsizing of mainframe applications and upsizing of microcomputer applications.

Client-Server Model

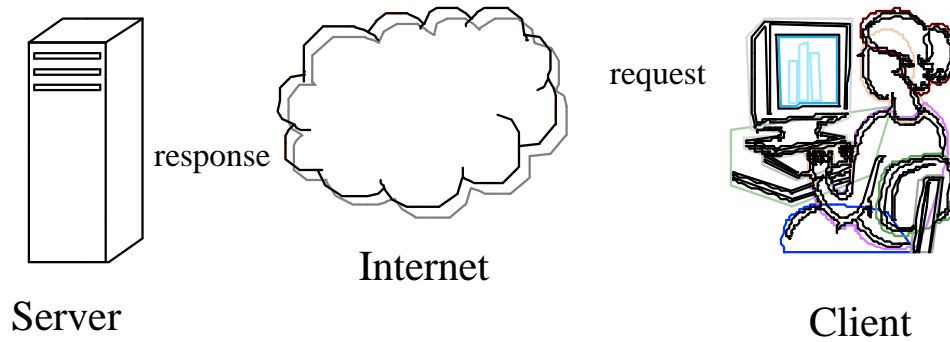


Figure 3.1: Client-server Model

The client requests some type of service (such as file or database access) from the server. The server fulfills the request and transmits the results to the client over a network. While both the client and the server programs can reside on the same computers, typically they run on different computers. Specific types of clients include web browsers, email clients, and online chat clients. Specific types of servers include web servers, ftp servers, application servers, database servers, name servers, mail servers, file servers, print servers, and terminal servers. Most web services are also types of servers. Where a server is made to handle request from multiple clients and transaction processing is done on both the server and the client we have distributed processing. Distributed processing involves the storage of data on database servers called back-ends from where clients' applications called front-ends access the data needed for their operations. In addition, the client (front-end) does data presentation and or processing, while the server (back-end) does storage, security and major data processing. The client / server inter-relationship is given in terms of layers and tiers. The Internet is a great example of client / server architecture at work. Consider a scenario where a user accesses his bank from a location away from the bank's computer. The user will require a web browser client to send a request to a web server at a bank. That program may in turn forward the request to its own database client program that sends a request to a database server at another bank computer to retrieve the account information. The balance is returned to the bank database client, which in turn sends it back to the web browser client displaying the results to the user. The client-server model has become one of the central ideas of network computing. Many business applications being written today use the client-server model. So do the Internet's main application protocols, such as HTTP, SMTP, Telnet, and DNS.

3.3 Types of Server

Servers are usually high-performance computers connected to the Internet by high speed communication lines. Depending on your application, you may deploy it on less-powered machine with less substantial connections. The following are variation of servers:

- a) Web server: This is use to store and deliver the elements of web pages.

- b) Application server: This is used to run specialized Internet application, such as ecommerce engine. It is designed to process requests and deliver dynamic results.
- c) Streaming server: This is used to deliver audio or video to the visitors to a site real-time.
- d) Mail server: This is used to send and receive email
- e) Name server: This is a specialized server that stores huge directories of web servers. It keeps track of all the registered domain names on the Internet
- f) Secure server: This is a Web server that encrypts data before transmitting it, to prevent unauthorized access. They are commonly used to secure for financial transactions in the Internet

3.4 Web Application

Webapps are applications that are accessed with a web browser over a network such as the Internet or an intranet. They are popular because of the ubiquity of the browser as a client (thin client). Similarly, the popularity is equally due to the possibility of updating and maintaining the application without necessarily distributing and installing it on every available client. Webapps or weblications as they are sometimes called are used to implement webmail, online retail sales, online auctions, discussion boards, weblogs etc. Web developers often use client-side scripting to add functionality to the webapps by creating an interactive site that does not require page reloading. Webapps generate a series of web pages dynamically in a standard format such as hypertext markup language (HTML) supported by common browsers. Through Java, JavaScript, Flash and other technologies, application specific methods such as drawing on the screen, playing audio and accessing the keyboard and mouse are all possible. Webapps are the present and the future of business transactions. The web works based on the client/server Architecture. That is, both the server and the client application are responsible for some sort of processing. Web application is commonly structured as a 3-tier application. The web browser constitutes the first tier, a middleware engine using some dynamic web content technology such as: common gateway interface (CGI), hypertext preprocessor (PHP), Java Servlets or Java server page (JSP), active server pages (ASP) constitute the middle-tier and the database is the third tier or back-end. The backend applications include MySQL, SQL Server, Oracle, etc. The bulk of online transactions take place between the middleware and the database server. While the middle-ware is responsible for the business logic transaction processing, the back-end is responsible for information storage and retrieval from the database.

The middle-tier may be multi-tiered. That is, it can be composed of several other servers with designated responsibilities, hence the over-all architecture is said to be N-tier. A fundamental rule in a 3-tier architecture is that the client has no direct line of communication with the data tier. That is, all communications are routed through the middleware tier.

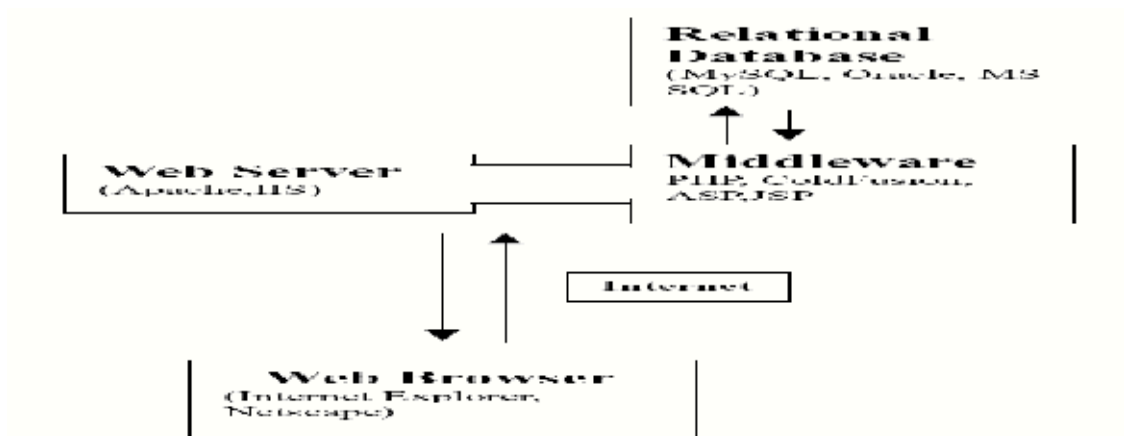


Figure 3.2: Architecture of a Web Application

3.4.1 The Web Browser

Client-side refers to operations that are performed by the client in a client–server relationship in a computer network. Typically, a client is a computer application, such as a web browser, that runs on a user's local computer or workstation and connects to a server as necessary. Operations may be performed client-side because they require access to information or functionality that is available on the client but not on the server. Programs that run on a user's local computer without ever sending or receiving data over a network are not considered clients, and so the operations of such programs would not be considered client-side operations. The web browser constitutes the client. It is a software application that enables a user to display and interact with text, images and other information that are located on the web page or a local area network. Browsers can be used to access information on web servers. Examples of web browsers are MS Internet Explorer, Mozilla Firefox, Apple Safari, Netscape and Opera and Google Chrome. Web browsers communicate with web servers using the hypertext transfer protocol (HTTP) to fetch web pages and it allows web browsers to submit information to web servers as well as fetch web pages from them. The primary language of browsers is the HTML, which consists of tags that are used to describe a web page. Most browsers have some level of support for Javascript and extensible markup language (XML).

3.4.2 The Web Server

A web server can be referred to as either the hardware (the computer) or the software (the computer application) that helps to deliver content that can be accessed through the Internet. The most common use of Web servers is to host Web sites but there are other uses like data storage or for running enterprise applications. The primary function of a web server is to deliver web pages on the request to clients. This means delivery of HTML documents and any additional content that may be included by a document, such as images, style sheets and JavaScripts. A client, commonly a web browser or web crawlers, initiates communication by making a request

for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. The resource is typically a real file on the server's secondary memory, but this is not necessarily the case and depends on how the web server is implemented. While the primary function is to serve content, a full implementation of HTTP also includes receiving content from clients. This feature is used for submitting web forms, including uploading of files. Many generic web servers support server-side scripting. The scripting tools used for middleware development include PHP, JSP, ASP, Servlet, PERL, python etc. These tools allow the behaviour of the web server to be scripted in separate files, while the actual server software remains unchanged. Usually, this function is used to create HTML documents "on-the-fly" as opposed to returning fixed documents. This is referred to as dynamic and static content respectively. The former is primarily used for retrieving and/or modifying information from databases. The latter is, however, typically much faster and more easily cached. Web servers are not only used for serving the world wide web, they can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring and/or administrating the device in question. This usually means that no additional software has to be installed on the client computer, since only a web browser is required (which now is included with most operating systems). There are many web server programs available. Table 3.1: Shows a statistics of the market share of the top servers on the Internet by Netcraft survey in March 2011.

Vendor	Product	Web Site Hosted	Percent
Apache	Apache	179,720,332	60.31
Microsoft	IIS	57,644,692	19.34
Igor Sysoev	nginx	22,806,060	7.65
Google	GWS	15,161,530	5.09
Lighttpd	lighttpd	1,796,471	0.60
Sun Microsystems	SunOne		

Table 3.1: Popular Servers

Servers are slave programs. They act only when request are made to them by browsers running on other computers and the Internet. The most commonly used Web Servers are Apache which has been implemented for variety of computer platforms, and Microsoft's Internet Information Server (IIS), which runs under windows operating systems.

1. Apache HTTP Server

This is most popular web server. It is a free software/open source like Linux, PHP and MySQL. Apache runs on Unix, Linux, MS Windows, Novell Netware and some other platforms. Apache serves over 68% of websites and serves both static and dynamic contents on the web in a very reliable and secure manner. Apache began as a the NACA server, httpd, with some added features. The name Apache has nothing to do with the Native American tribe of the same name. Rather, it came from the nature of its first version, which was patchy version of the httpd server. As seen in the usage statistics it is the most widely used server. The primary reasons for this are as follows: It is an excellent server because it's both fast and reliable. Furthermore, it is open-

sources software, which means it is free and managed by a large team of volunteers, a process that efficiently and effectively maintains the systems. Finally, it is one of the best available for Unix-based systems.

2. The Internet information services (IIS)

The Internet information services (IIS) is a server or system based services for servers using Microsoft Windows operation system. It is a major component of the Microsoft Server operating system and particularly, a component of its active server pages (ASPs). IIS is recommended if both the middleware (ASP) and the database Server (SQL Server) are Microsoft products. Though the Apache server may be installed on Windows platforms, it is not the most popular server on those systems. IIS remains the most popular on Windows platform because it is supplied as part of Windows and also because it is a reasonable good server. Apache and IIS provide similar varieties of services.

In summary you can distinguish between Web Client and Web Servers as follows:

Web Client

- Connected to the Internet when needed
- Usually runs Web browser(client) software such as Internet Explorer or Netscape
- Uses HTTP
- Request Web pages from a server
- Receives Web Pages and files from as server

Web Server

- Continually connected to the Internet
- Runs Web server software (such as Apache or Internet Information Services (IIS))
- Uses HTTP
- Receives a request for the Web page
- Responds to the request and transmits the status code, Web page, and associated files

3.5 OSI Reference Model Concept

Another model, the Open Systems Interconnection, or OSI, model was designed by the International Organization for Standardization (ISO). It is a seven-layered model. OSI was never seriously implemented as a protocol stack, however, it is a theoretical model designed to show how protocols stack should be implemented. The OSI model simplifies complex networking activities by grouping the steps in the process into seven separate task layers (The physical, Data Link, Network, Transport, Session, Presentation, and Application layer). This is shown in Table 3.2 below. By dividing the process into smaller tasks, it becomes easy for vendors to manage smaller pieces of the problem.

Number	Name	Function

Layer 1 Physical	This layers consist of the	networking media (wiring and interconnections) and the components necessary to transmit a signal from one end to the other
Layer 2 Data Link	Layer This layer packages	the data so that it can be transmitted over the Physical layer
Layer 3 Network Layer	This layer is where data	is separated into frames. It also determines the route the data will take to the destination
Layer 4 Transport	This layer ensures data packets	are sequenced properly and do not contain any errors
Layer 5 Session	This layer maintains a connected	link, called a session, between the two communication ends
Layer 6 Presentation	layer It determines the	format used for communication and compresses, encrypts, or converts the data as necessary for the protocol in use
Layer 7 Application	Layer This layer completes	or initiates the actions being communicated.

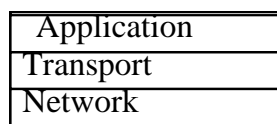
Table 3.2: OSI model

Although the seven layers of the OSI model describe unique tasks performed during network communications, the demarcation between the layers and the total number of layers is irrelevant as long as all the actions are accomplished. Several other models exist apart from the OSI model.

3.6 TCP/IP Protocols Suit

A new model with only five layers would be a bit easier to understand than the OSI model. Table 3.3 presents the layered protocol stack that dominates data communications and networking today. It is a five-layered Internet model sometimes called the TCP/IP protocol suite. The model is composed of five ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), and application (layer 5). Each layer defines a family of functions distinct from those of other layers.

Within a single machine, each layer calls upon the services of the layer just below it. Layer 4, for example, uses the services provided by layer 3 and provides services for layer 5. Between machines, layer y on one machine communicates with the corresponding layer y on another machine. This communication is governed by an agreed-upon series of rules and conventions called protocols. The processes on each machine that communicate at a given layer are called peer-to-peer processes. Thus, the communication between machines is therefore a peer-to-peer process using appropriate protocols for a given layer.



Data Link
Physical

Table 3.3: Internet Model

3.7 COMMON GATEWAY INTERFACE (CGI)

Most users of the Internet would always prefer to visit sites that are responsive and interactive sites. Common Gateway Interface is a standard that allows you add dynamic functionalities to static web pages. Dynamic functionality is a requirement for site that implements site search, order form, e-mail, database display, or other type of processing. These applications expect responses from a server based on client requests. This is when server-side processing is needed, commonly known as the Common gateway Interface (CGI). CGI is a thus a standard that defines how Web server software can delegate the generation of web pages to a client application or an executable file. It is a technology that creates and handles dynamic documents. CGI defines how a dynamic document should be written, how input data should be supplied to the program, and how the output result should be used. The use of 'common' in CGI connotes that the standard defines a set of rules or protocols that are common to any language or platform. The 'gateway' means that a CGI program is a gateway that can be used to access other resources such as databases and graphic packages. While the term 'interface' implies that there is a set of predefined terms, variables, calls, etc that can be used in any CGI program. CGI scripts preferably written in scripting languages such as PHP, ASP, JSP or PERL though a common programming language such as C can also be used.

CGI programs

A CGI program in its simplest form is code written in one of the languages supporting CGI. Once you can encode a sequence of thoughts in a program and you are familiar with the syntax of one of the above-mentioned languages, writing simple CGI programs becomes very easy. The program in example 1 below is used to outputs the systems date to the browser.

A CGI program written in HTML

Example 1

```
#!/bin/sh
#The head of the program
echo Content_type:text/html
echo
# The body of the program
echo <HTML>
echo <HEAD><TITLE> Date and Time </TITLE></HEAD>
echo <BODY>
now=`date`
echo <CENTER><B>$now </B></CENTER>
echo</BODY>
echo</HTML>
```

exit 0

Though example 1 is used to demonstrate how a simple CGI program could be written, CGI is a standard method, for a Web server to pass a Web page user's request. This is usually initiated through the use of a form element in HTML to an application program and to accept information to send to the user. As soon as the Web server receives the request (as detailed in the form) it passes the form information to a small application program that processes the data, and sends back a confirmation Web page or message to the browser. This specification for passing data back and forth between the server and the application is called CGI and is part of the Hypertext Transfer Protocol (HTTP) specification.

Active Document

The program that runs from the client side is known as the active document. This is the document or page that may make a request to the server. For example, imagine you want to complete a registration form at National Open University's site. This program would certainly need to run at the client (user's) site where the online form is made available. When a browser requests an active document, the server sends a copy of the document in the form of byte code. The document is then run at the client (browser) site. An active document is stored in the server in the form of binary code. However, it does not create any form of overhead for the server in the same way that a dynamic document does. When a client retrieves an active document from a server it may store it in its storage area so that should there be a need for re use, it wouldn't need to make another request before it can be used. An active document is transported from the server to the client in binary form. The compression of an active document at the server side and its decompression at the client side helps to save bandwidth and transmission time during transportation.

Dynamic Document

Dynamic documents do not exist in a predefined format instead they are created by a Web server whenever a browser requests the document. When a web browser requests Web pages and their related files from a web server, the Web server locates the files and sends them to your Web browser. Then the Web browser renders the returned files and displays the requested Web pages. Because a fresh document is created for each request, the contents of dynamic document can vary from one request to another. For example, imagine you are to get the systems time and date from a server at different times. Time and date values are kinds of information that are dynamic in that they change from moment to moment. At different instances of such calls one expects to receive different values for time delivered to the client by the server.

CGI – Server Side Processing

A Web page engages CGI by either an action attribute on a form or a hyperlink. At this instance any form data that exists is passed to the CGI script. The CGI script usually written in PHP, ASP, JSP, ColdFusion or PERL would complete the processing and may create a confirmation or response message which is rendered to the browser (client). Have you considered that any time you use Google or other search engine, you are really using CGI.

Steps in Utilizing CGI

Web page engages CGI by a form or hyperlinks
 Web server executes server-side script or program
 Server-side script accesses requested database(where available), file or process
 Web server returns Web page with requested information or confirmation of action to the browser.

The location of the script being used must be known for proper referencing especially if a third party is involved in Web application development.

4.0 Conclusion

The client / server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often this communication is over a computer network on separate hardware, but both client and server may reside in the same system. In this section we covered the network model and protocols that make web application development possible.

5.0 SUMMARY

A server machine is a host that is running one or more server programs which share resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests. This specification for passing data back and forth between the server and the client application is called CGI and is part of the Hypertext Transfer Protocol (HTTP) specification. CGI scripts are written in PHP, ASP, JSP, ColdFusion, PERL, C or any other programming languages.

6.0 TUTOR MARKED ASSIGNMENTS

- a) With the aid of a diagram explain the term Web application
- b) Discuss the OSI Reference Model Concept
- c) What is a Web Server ? Give two examples
- d) What are the steps in utilizing CGI?

7.0 REFERENCES / FURTHER READINGS

Alex Leon and Mathew Leon (1999), Fundamentals of Information Technology, Vikas Publishing House PVT LTD, New Delhi

Andy Sloane (1999), Computer Communications, Principles and Business Applications, Second Edition, McGraw-Hill, England

Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Brian K. Williams, Stacey C. Sawyer, Sarah E. Hustchinson(1999) Using Information Technology: A Practical Introduction to Computers and Communication, Irwin/McGraw-Hill

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

Katarzyna J. Macura,Internet and You: Connecting to the Internet

<http://www.rsna.org/Technology/internet2-1.cfm>

[http://books.mcgraw-hill.com/downloads/products/0072194154/007](http://books.mcgraw-hill.com/downloads/products/0072194154/0072194154.ch01.pdf)

[2194154.ch01.pdf](http://books.mcgraw-hill.com/downloads/products/0072194154/0072194154.ch01.pdf)

<http://www.wisegeek.com/what-is-a-web-browser.htm>

MODULE TWO

UNIT 1: UNDERSTANDING HTML

CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES

3.0 MAIN CONTENT

3.1 Background of HTML

3.2 Browser

3.3 HTML Tags

3.4 The structure of WEB Page

3.5 How to Create and Run HTML codes

4.0 CONCLUSION

5.0 SUMMARY

6.0 TUTOR MARKED ASSIGNMENT

7.0 REFERENCES / FURTHER READINGS

1.0 Introduction:

HTML stands for Hypertext Mark-up Language. It is the language for building Web pages and consists of standardized codes or tags that are used to define the structure of information on the Web page. Web pages come in many different varieties. In their simplest form, they contain information which is made up of simple texts. On the other extreme are pages, which are highly colourful, containing animation, sound and interactive elements. HTML codes makes it possible for web pages to have many features including bold text, italic text, heading, paragraph breaks, tables, forms etc. Web pages generally reside on the HTTP server. A user request a web page from an HTTP (web) server through a web browser such as, Internet Explorer, Mozilla Firefox, Safari, Chrome etc, either by clicking on the hypertext or designating a particular URL (Uniform Resource Locator). The server then sends the requested information to the user computer.

In this unit we will discuss the background of HTML, Web browsers, tags and editors.

2.0 Objectives

At the end of this unit, the students will be able:

- to discuss the meaning of HTML
- to understand simple HTML codes
- to use Web browsers to display HTML codes
- to write HTML codes using popular tags
- to run HTML codes

3.0 Main Content

3.1 Background of HTML

HTML is the set of mark-up symbols or codes placed in a file intended for display on the Web browser page. These mark-up symbols and codes identify structural elements such as paragraphs, heading, and lists. HTML can be used to place media (such as graphics, video, and audio) on the Web page and describe fill-in-forms. The browser interprets the mark-up codes and renders the page. HTML permits the platform-independent display of information across network. That is, no matter what type of computer a Web page was created on, any browser running on any operating system can display the page. The newest version of HTML used today is eXtensible HyperText Mark-up Language (XHTML). XHTML uses the tags and attributes of HTML along with the syntax of eXtensible Mark-up Language (XML). Our focus is on HTML

3.2 Browser

A Web browser is a software program that interprets the coding language of the World Wide Web in graphic form, displaying the translation rather than the coding. A browser acts as an interface between the user and the inner-working of the web. The browser software such as the Internet Explorer, Firefox, Mozilla, Safari, etc interpret HTML codes and presents the information contained in the web pages in a readable format on the users' computer. A browser does not display HTML tags. Browsers function as client programs by contacting the web server and sending the request for information received to the users' computers.

3.3 HTML Tags

HTML consists of standardized tags that are used to define the structure of information on the Web pages. The decision about the structure of the text is made by the browser based on the tags which are marks that are embedded into the text. A tag is enclosed in two signs (< and >) and usually comes in pairs. The beginning tag starts with the name of the tag, and the ending tag starts with a slash followed by the name of the tag. The use of tags enables web pages to have many features including bold text, italic text, heading, paragraph breaks and numbered or bulleted list. Table 2.1 shows a list of common HTML tags

Beginning Tag	Ending Tag	Meaning
<A>		Defines an address (hyperlink)
<BODY>	</BODY>	Defines the body of the document
 		Line break
<HEAD>	</HEAD>	Defines the head of the document
<HN>	</HN>	Defines different Headers (n is an integer)
<HTML>	</HTML>	Defines an HTML document
		Define an Image
		An item in a list
		Ordered list
<TITLE>	</TITLE>	Defines the title of the document

Table 2.1: HTML tags

Tags are generally used to specify "mark-up" regions of HTML documents for the web browser to interpret. Tags are composed of the name of the element, surrounded by angle brackets. An

end tag also has a slash after the opening angle bracket, to distinguish it from the start tag. For example, p which is represented by p element, would be written as

```
<p> This is my first HTML code ....</p>
```

Not all elements require the end tag. An example of an element that does not require an end tag is the
 element which forces a line break on the display of interpreted HTML codes on a browser.

HTML attributes are modifiers of HTML elements. They generally appear as name-value pairs, separated by "=", and are written within the start tag of an element, after the element's name:

```
<"tag" "attribute"="value">(content to be modified by the tag)</tag>
```

Where tag names the HTML element, attribute is the name of the attribute, set to the provided value. An attribute customizes or modifies HTML elements.

3.4 The structure of WEB Page

The basic structure for all HTML documents is simple and should include the following minimum elements or tags:

```
<html>-This is the main container for HTML pages  
<head>-This is the container for page header information  
<title>-The is used for the title of the page  
<body>-This is a container of the main body of the page
```

The <HTML> Element

The HTML element is considered the root and container element for the whole HTML document. That is, its sole purpose is to encapsulate all the HTML code and describe the HTML document to the web browser. Each HTML document should have one <html> and each document should end with a closing </html> tag.

Example 1: HTML Code:

```
<html>  
</html>
```

The <HEAD> Element

The HEAD tag marks the beginning of the document head element and contains the title of the pages and other parameters that the browser will use. Thus Each <head> element should contain a <title> element indicating the title of the document, and may also contain any combination of the following elements, in any order:

The <style> tag.

This is used for declaring or including Cascading Style Sheets(CSS) codes inside your HTML document.

The <script> tag

This tag is used to declare or include JAVAScript or VBScript inside the document.

The <meta> tag

This is used to include information about the document such as keywords and a description, which are particularly helpful for search applications.

The <base> tag

This is used to create a "base" universal resource location(url) for all links on the page.

The <object> tag

This is designed to include multimedia such as images, Flash animations, MP3 files, QuickTime movies JavaScript objects, and other components of a page. The <param> tag is used along with this tag to define various parameters. Note the <embed> tag can also be used to include multimedia files as will be discuss later in this module.

The <link> tag

This is used to link to an external file, such as a style sheet or JavaScript file.

Example 2: Codes for HEAD element

```
<head>
<meta name="Keywords" content="NOUN, Web Pages" />
<meta name="description" content="HTML Basic Tags" />
<base href="http://www.nou.edu.ng " />
<link rel="stylesheet" type="text/css" href="noun.css" />
<script type="text/javascript">
_uacct = "UA-232293";
urchinTracker();
</script>
</head>
```

The <title> Element:

The <title> tag is usually placed within the <head> element to title your page. Whatever is written between the opening and closing <title></title> tags will be displayed in the title bar of the WEB browser. The title information is used by search engines that use its content to help index pages. Therefore it is excellent practice to use a title that really describes the content of your site.

Example 3: code for TITLE element

```
<html>
<head>
<title>National Open University of Nigeria </title>
</head>
</html>
```

The <Body> Element

The <body> element appears after the <head> element. The purpose of the <body> element is to contain the text and HTML element that will display in the browser window. A <body> element may contain anything from a couple of paragraphs, links, images under a heading to complicated layouts containing forms and tables. We will be looking at each of these elements in greater detail later in this unit. For now, it is only important to understand that the body element will encapsulate all of your webpage's viewable content.

Example 4: Codes for BODY element

```
<html>
<head>
<title>National Open University Website!</title>
</head>
<body>
Welcome to the official Website of the National Open University of Nigeria
</body>
</html>
```

Example 5: Codes for HTML, HEAD, TITLE and BODY tags put together

By putting all the tags together we have a complete HTML document as follows:

```
<html>
<head>
<title>National Open University of Nigeria </title>
<meta name="Keywords" content="NOUN, Web Pages" />
```

```
<meta name="description" content="HTML Basic Tags" />
<base href="http://www.nou.edu.ng " />
<link rel="stylesheet" type="text/css" href="noun.css" />
<script type="text/javascript">
_uacct = "UA-232293";
urchinTracker();
</script>
</head>
<body>
  <P>Welcome to the official Website of the National Open University of Nigeria.</p>
</body>
</html>
```

3.5 How to Create and Run HTML codes

Creating an HTML document is easy. HTML lets us use only ASCII characters for both the main text and formatting instructions. To begin coding HTML you need a standard text editor. Notepad is readily available on computer systems. You could also use Dreamweaver or other text editors if installed. It is not advisable to use a word processor.

Steps

1. Launch Notepad application on your computer
2. Type in your HTML codes
3. Save the document onto a location in your computer drive with a name and the extension
html [e.g firstnoun.html]
4. Click on the filename created.

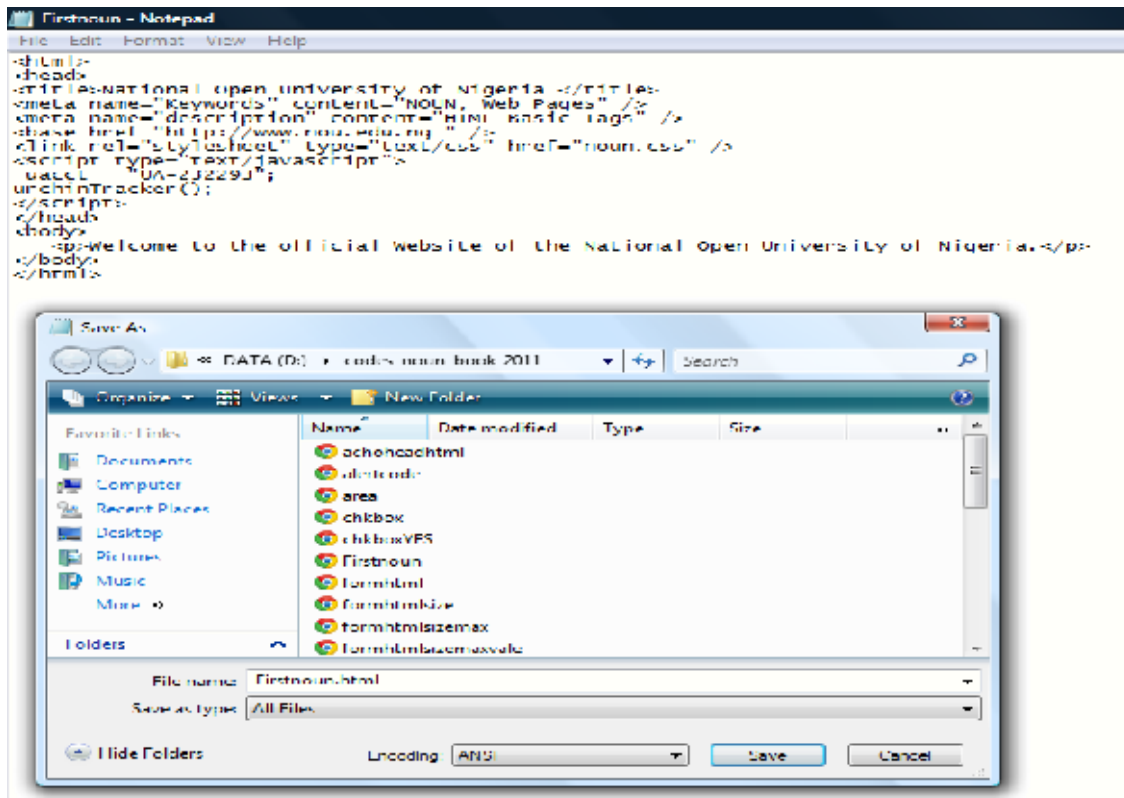


Figure 1: Step 1-3 of Section 3.5

Practice 1

Type in the HTML Codes in Example 5 and follow the steps in section 3.5. Run the codes by clicking on the filename and see how it appears

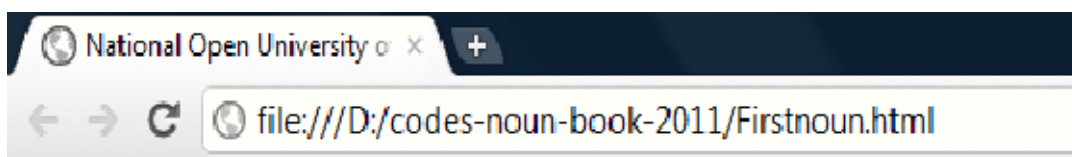
Answer to Practice Question 1

Figure 2: Display interpreted HTML code in a Google Chrome browser

4.0 CONCLUSION

HTML uses tags which allow symbols or codes placed within the opening and closing tags to be rendered to a browser. Examples of browsers include Internet Explorer, Google Chrome, Firefox etc. Browser interprets mark-up codes and renders the page to display. HTML permits the platform-independent display of information across network. HTML can be used to place media (such as graphics, video, and audio) on the Web page and also to describe fill-in-forms.

5.0 SUMMARY

The development of Web pages is an interesting task. In this unit we have been able to lay a good foundation for designing simple and complex web pages. We can now identify HTML tags, put a few of them together and get the browser to interpret them.

6.0 TUTOR MARKED ASSIGNMENT

1. What is the meaning of HTML?
2. List and explain the meaning of 5 HTML tags
3. List the name of 5 Web browsers
4. Write HTML codes to display your personal details to include name, address, department and year of study

7.0 REFERENCES / FURTHER READINGS

Alex Leon and Mathew Leon (1999), Fundamentals of Information Technology Vikas Publishing House PVT LTD, New Delhi,

Miller David (2006), Data Communications and Network , McGraw Hill, N.Y, 2006

Terry Felke-Morris(), Web Development and Design Foundations with XHTML, Pearson International Edition

<http://www.wisegeek.com/what-is-a-web-browser.htm>

UNIT 2: HTML ELEMENTS

CONTENTS

- 1.0 INTRODUCTION
- 2.0 OBJECTIVES
- 3.0 MAIN CONTENT
 - 3.1 Presentational Tags:
 - 3.2 Multimedia Elements
 - 3.2.1 GIF Image File Format
 - 3.2.2 The JPEG Image File Format
 - 3.2.3 The PNG Image File Format
 - 3.3 Image Element
 - 3.4 HTML Music (audio) and Video Codes
 - 3.5 Anchor Element
 - 3.6 HTML Email Tag
- 4.0 CONCLUSION
- 5.0 SUMMARY
- 6.0 TUTOR MARKED ASSIGNMENT
- 7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

Just as in word processor software, you would want to make text bold, italicized, or underlined and apply other forms of formatting to your texts. With HTML these are also possible with the tags that will be discussed in this unit. In addition, we shall also cover how to embed multimedia (images, audio, and video) elements in Web pages.

2.0 OBJECTIVES

At the end of this unit the student should be able to:

- write HTML codes that enhance the appearance of web pages
- identify multimedia elements
- describe different multimedia formats
- embed multimedia files in HTML codes
- create link to other web pages

3.1 Presentational Tags:

Formatting is possible in HTML with the with the following tags.

Element	Example	Usage
	Bold Text	Display text in bold
<i>	Italic text	Display text in italic
<big>	Big text	Display text larger than their normal size

<small>	Small text	Display text smaller than normal size
<center>	</CENTER>	Centred
<sub>	Subscript Text	Display small text above the baseline
<sup>	Superscript text	Display small text below the baseline
<strike>	Strike	Displays text with a line through it
<u>	U text	Italic

The Paragraph Elements

Paragraph elements are used to group sentences and sections of text together. Texts to appear in the paragraph are contained between <p> and </p> tags.

```
<p> National Open University of Nigeria.</p>
<p> 14/16 Ahmadu Bello Way, Victoria Island </p>
<p> Lagos Nigeria.</p>
```

This will produce the following result:

National Open University of Nigeria
14/16 Ahmadu Bello Way, Victoria Island
Lagos Nigeria.

The line Break Element

The line break tag,
, is used to force a new line when the text on the Web page document is displayed by a browser. The line break tag opening tag is used alone. It is not used as in pairs like many other tags.

```
<br> National Open University of Nigeria.
```

This will produce the following result:

Example 1: HTML Codes: Putting all Together

```
<html>
<head>
<body>
<p><b> National Open University of Nigeria </b>
<br><i> National Open University of Nigeria</i>
<br><big> National Open University of Nigeria</big>
<br><small> National Open University of Nigeria</small>
<br><center> National Open University of Nigeria</center>
<br><sub> National Open University of Nigeria</sub>
<br><sup> National Open University of Nigeria</sup>
<br><strike> National Open University of Nigeria</strike>
<br><u> National Open University of Nigeria</u>
```

```

</p>
</body>
</head>
</html>

```

This will produce:

National Open University of Nigeria
National Open University of Nigeria
 National Open University of Nigeria
 National Open University of Nigeria

National Open University of Nigeria

National Open University of Nigeria
 National Open University of Nigeria
~~National Open University of Nigeria~~
National Open University of Nigeria

The Heading Elements

These are organized into levels <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. The size of the text is largest at <h1> and smallest for <h6>. Depending on the font being used, text contained in <h5> and <h6> may be displayed smaller than the default text size. You use different sizes for your headings. When headings are displayed by a browser, one line is added before and after each of the heading

Example 4: Codes for Heading element

```

<html>
<head>
<body>
<h1>Heading1: National Open University of Nigeria </h1>
<h2> Heading2: National Open University of Nigeria </h2>
<h3> Heading3: National Open University of Nigeria </h3>
<h4> Heading4: National Open University of Nigeria </h4>
<h5> Heading5: National Open University of Nigeria </h5>
<h6> Heading6: National Open University of Nigeria </h6>
</body>
</head>
</html>

```

This will display following result:

Heading1: National Open University of Nigeria

Heading2: National Open University of Nigeria

Heading3: National Open University of Nigeria

Heading4: National Open University of Nigeria

Heading5: National Open University of Nigeria

Heading6: National Open University of Nigeria

3.2 Multimedia Elements

Multimedia is media and content that uses a combination of different content forms. It includes a combination of text, audio, images, animation, video, and interactivity content forms. Multimedia elements such as sounds or videos are stored in media files. Multimedia comes in many different formats and are usually embedded in web pages to enhance their appearance and interactivity. Most web browsers today have support for a number of multimedia formats. The ones that are commonly supported are are: CompuServe Graphic Format (GIF), Joint Photographic Expert Group (JPEG or JPG) , and Portable Network Graphics (PNG) .

3.2.1 GIF Image File Format

The Graphics Interchange Format (GIF) is a bitmap image format bitmap image format that was introduced by CompuServe in 1987 and has since come into widespread usage on the World Wide Web due to its wide support and portability. It is best used for banners, clip art, and icons. The main reason for this is that gifs can have a transparent background which is important for web design. GIF files are usually larger files, which lead to low download times and large transfer rates. Gifs are also limited to the 256 color scheme. The color limitation makes the GIF format unsuitable for reproducing color photographs and other images with continuous color, but it is well-suited for simpler images such as graphics or logos with solid areas of color.

3.2.2 The JPEG Image File Format

The Joint Photographic Experts Group (JPEG) developed the JPEG image file format. As the name implies, it is suitable for photographic images containing many colors, as supports over 16.7 million colors (with 24-bit color). They are easier to download than gif s files and saves hard drive space because of the support of high compression. It is best to use Jpegs for photo galleries, or artwork to allow the viewer to catch that extra bit of detail

3.2.3 The PNG Image File Format

The Portable Network Graphics (PNG - pronounced ping) image file format was developed in response to sever al needs such as more efficient format and the compression algorithm patent

controversy associated with GIF format. It was created as the free, open-source successor to the GIF. It combines the best of GIF and JPEG. The PNG file format supports true color (16 million colors) while the GIF supports only 256 colors. The PNG file excels when the image has large, uniformly colored areas

3.3 Image Element

Images can be included within HTML documents with the IMG element tag . Images are used to enhance the appearance of web pages. Commonly used attributes with the image element are shown in table

Attribute	Value	Description
Alt Text	Specifies	an alternative text for an image
Src URL	Specifies	the URL or source of an Image

Image src

From the above Src attribute stands for source, i.e the source where the image file is located. The image may be directly available on your local system, a web server or any standard URL. The src attribute must be properly pointed to the local or external source.

Example : HTML Code

```

```

Image:



HTML –Alternative Attribute

The alt attribute specifies alternate text to be displayed if for any reason, the browser does not locate the specified image or if a user has image files disabled. For browsers that support only text, the alternate attribute becomes very relevant also.

Example HTML Code:

```

```

3.4 HTML Music (audio) and Video Codes

Music and Video can easily be inserted onto web page in a relatively easy way by using the embed <embed> tags. In the past multiple tags had to be used because browsers did not have a uniform standard for defining embedded media files. A src attribute is required to define the correct URL (local or global) of the audio or video file in order for it to be displayed correctly. Other attributes can be set in order to customize your web pages.

Example :HTML Cod for Music (audio)

```
<embed src="nounantheme.mid" />
```

Example HTML Code for Video

```
<embed src="http://universitymedia.com/files/noun.mpeg"/>
```

Listing of Video Media Types

The list below is the most commonly used file formats for the internet.

- .swf files - are the file types created by Macromedia's Flash program.
- .wmv files - are Microsoft's Windows Media Video file types.
- .mov files - are Apple's Quick Time Movie format.
- .mpeg files - set the standard for compression movie files created by the Moving Pictures Expert Group.

Flash movies (.swf), AVI's (.avi), and MOV's (.mov) file types are supported by the embed tag. The .mpeg" files and Macromedia's .swf files are the most compact and widely used for the design of web pages.

3.5 Anchor Element

The HTML anchor <a> element is used to create a link or hyperlink reference (href) to a resource such as another web page, a file, a multimedia element, etc. or to a specific place within a web page. Each hyperlink begins with an <a> tag and ends with an tag. The opening and closing tags surround the text to click to perform hyperlink. The anchor tag requires some attribute and value to work.

Hypertext Reference (href)

The href attribute defines reference that the link refers to. Basically this is where the user will be taken if they wish to click this link. Hypertext references can be Internal, Local, or Global.

Example


```
<a href="http://nou.edu.ng">NOUN</a>
<a href="home.html">Home</a>
<a href="contactus.html">Contact Us</a>
<a href="register.html">Register</a>
<a href="login.html">Login</a>
```

In this example when you click on NOUN, it will take you to the website nou.edu.ng

3.6 HTML Email Tag

To create email link use a standard HTML anchor tag `<a>` and set the `href` property equal to the email address, rather than specifying a web URL.

Example HTML Code:

```
<a href="mailto:myaddress@gmail.com">Email </a>
```

4.0 CONCLUSION

The appearance of Web pages goes a long way in determining the number of visitors and how long they spend in a site. It is therefore necessary for Web developers to be conversant with how to use relevant HTML tags for enhancing the presentation of text and display of images in Web Pages. Many of these tags have been covered in this unit. The three most common methods of representing images are Graphic Interface For mat (GIF), Joint Photographic Experts Group (JPEG) and Portable Network Graphics (PNG). These formats allow images to be compressed for easy download from Web pages.

5.0 SUMMARY

This unit described how the text content of an HTML document can be formatted with relevant HTML tags. By formatting we mean layout and some presentation details. We also discussed how multimedia elements can be embedded into HTML documents to enhance their interactivity or appearance.

6.0 TUTOR MARKED ASSIGNMENT

1. Describe three types of Image Format
2. List and explain the meaning of 5 HTML tags
3. List five (5) popular Web browsers
4. Write HTML codes to display your personal details to include name, address, department and year of study

7.0 REFERENCES / FURTHER READINGS

Barbara K. Kaye, Norman J. Medoff (2001) *The World Wide Web: A Mass Communication Perspective*, Mayfield Publishing Company, USA

Barrie Sosinsky, Valda Hilley, (2004) *Programming the Web: An Introduction*, McGraw-Hill, USA

Behrouz A Forouzan (2003), *Data Communications and Networking*, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Deitel P. J., and Deitel H.M (2008), *Internet and World Wide Web: How to Program*, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

June Cohen (2003), *The Unusually Useful Web Book*, New Rider, USA

Marc D. Miller, Thomas C. Padgett (2003) *Web Design Using Macromedia Dreamweaver*, McGraw-Hill/ Irwin, USA

Peter Weverka (2001) *Instant Web Pages*, Sybex Inc, USA

Raymond Greenlaw and Ellen Hepp (2001), *In-Line/On-Line Fundamentals of the Internet and World Wide Web*, McGraw-Hill Learning Center, USA

Terry Felke-Morris (2009), *Web Development and Design Foundations with XHTML*, Pearson International Edition USA

UNIT 3: TABLES**CONTENTS****1.0 INTRODUCTION****2.0 OBJECTIVES****3.0 MAIN CONTENT****3.1 HTML Tables****3.2 Table Attributes****3.2.1 The border attribute****3.2.2 Spanning Multiple Rows and Cell****3.2.3 Table Heading – The <th> Element****3.2.4 CellPadding and Spacing****3.2.5 Colspan and Rowspan Attributes****3.2.6 The Width and Height Attributes****4.0 CONCLUSION****5.0 SUMMARY****6.0 TUTOR MARKED ASSIGNMENT****7.0 REFERENCES / FURTHER READINGS****1.0 Introduction**

A table is layout mechanism that allows arrange content which include text, images, links, forms, and even other tables to be arranged into vertical columns and horizontal rows. The rows and columns do not have to be visible, nor do they have to be equal in size. Each block of space within a table is called a table data cell. The size, background colors or background images, border colors and other formatting can easily be achieved by using Cascading Style Sheet In HTML, you create tables using the table tag, in conjunction with the tr and td as the basic tags. Major formatting can be done on table using CSS

2.0 Objectives

At this end of this unit, the student should be able:

- to design tables organize information using a table
- to create tables using table tags
- to format table by using appropriate table attributes

3.0 Main Content**3.1 HTML Tables**

The <table> tag is used to begin a table. Within a table element are the <tr> (table rows) and <td> (table columns) tags. Tables are can be used to create site layout and server as container for forms or other elements. To create a simple table of 2 rows and two columns the following HTML code can be used.

HTML Code:

```
<table border="1">
<tr><td>Row 1 Col 1</td><td>Row 1 Col 2</td></tr>
<tr><td>Row 2 Col 1</td><td>Row 2 Col 2</td></tr>
```

</table>

Basic Table:

Row 1 Col 1	Row 1 Col 2
Row 2 Col 1	Row 2 Col 2

Table rows are defined by <tr> and </tr> while table data is defined by <td> and </td>.

Let us now examine some attributes that are commonly used with tables

3.2 Table Attributes

Commonly used table <table> attributes include align, border, border color, with, height, cellpadding, cellspacing, bgcolor

3.2.1 The border attribute

This attribute is used to specify whether and what type of visible border the table will have. The values specifies range from 0 to 100 with 0 indicating no border will be visible and 1 (relatively thin border) -100 (very thick border). In the HTML code above we specified the value of the border attribute as 1.

3.2.2 Spanning Multiple Rows and Cell

The rowspan attribute is used to span multiple rows and while the colspan is used to span multiple columns. To set headers table you need the <th> tags. Headers are by default bold to make them different from other content of the table.

HTML Code:

```
<table border="1">
<tr>
<th>Column1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Col 1</td>
<td>Row 1 Column2</td><td>Row 1 Column3</td></tr>
<tr><td>Row 2 Column2</td><td>Row 2 Column3</td></tr>
<tr><td colspan="3">Row 3 Column1</td></tr>
</table>
```

Colspan and Rowspan:

Column 1	Column 2	Column 3
Row 1 Column1	Row 1 Column2	Row 1 Column3

	Row 2 Column2	Row 2 Column3
Row 3 Column1		

3.2.3 Table Heading – The <th> Element

Table heading can be defined using <th> element. The <th>, or table heading element is used to add heading to tables and distinguish column heading from table content. Figure 3.1 shows a table that uses <td> element

```
<table border="1">
<tr>
<th>Name</th>
<th>Department</th>
</tr>
<tr>
<td>Okeke Ayo Sule</td>
<td>Economics</td>
</tr>
<tr>
<td>Etinosa Wada Erujeje </td>
<td>Computer Science</td>
</tr>
</table>
```

This will produce following result. You can see its making heading as a bold one:

Name	Department
Okeke Ayo Sule	Economics
Etinosa Wada Erujeje	Computer Science

Figure 3.1: Table

3.2.4 CellPadding and Spacing

The cellspacing Attribute . This attribute specifies the distance between the cells in pixels. If you do not specify a value for the cellspacing attribute, the default value (usually around 2 pixel) is assumed by the browser.

The cellpadding attribute. This attribute specifies the distance in pixel between the cell content and the edge of the cell. If you do not specify a value for the cellpadding attributes, the default value which is 1 pixel is assumed by the browser. An example with cellpadding set to 10 is shown in Figure 3.2

HTML Code:

```
<table border="1" cellspacing="10">
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
<tr><td>Row 1 Col1</td><td>Row 1 Col2</td></tr>
<tr><td>Row 2 Col1</td><td>Row 2 Col2</td></tr>
</table>
```

Cellspacing and Padding:

Column 1	Column 2
Row 1 Col1	Row 1 Col2
Row 2 Col1	Row 2 Col2

Figure 3.2a: Cellspacing

Let us now change specify a value of 10 for the cellpadding of the table and remove the cellspacing from the previous example.

HTML Code:

```
<table border="1" cellpadding="10">
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
<tr><td>Row 1 Column1</td><td>Row 1 Column2</td></tr>
<tr><td>Row 2 Column1</td><td>Row 2 Column2</td></tr>
</table>
```

ColPads:

Column 1	Column 2
Row 1 Column1	Row 1 Column2
Row 2 Column1	Row 2 Column2

Figure 3.2b: Padding

The value you specify for padding and spacing is interpreted by the browser as a pixel value you. The value 10 specified in the two examples are simply means 10 pixels wide. That is, attributes that use numeric values for their measurements use pixels.

Figure 3.3 shows a table with cellpadding of 5 and cellspacing of 5.

```
<table border="1" cellpadding="5" cellspacing="5">
<tr>
<th>Name</th>
<th>Department</th>
</tr>
<tr>
<td> Okeke Ayo Sule </td>
<td>Economics</td>
</tr>
<tr>
<td> Etinosa Wada Erujeje </td>
<td>Computer Science</td>
</tr>

</table>
```

This will produce following result:

Name	Department
Okeke Ayo Sule	Economics
Etinosa Wada Erujeje	Computer Science

Figure 3.3: Cellspacing and Padding

3.2.5 Colspan and Rowspan Attributes

The colspan Attribute This attribute specifies the number of columns that a cell will occupy. It is also used to merge two or more columns into a single column.

The rowspan attribute. This attribute specifies the number of rows that a cell will occupy. The *rowspan attribute is used if you want to merge two or more rows.*

Figure 3.3 shows a table with rowspans=2 and colspan=3

```
<table border="1">
```

```

<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Column1</td>
<td>Row 1 Column2</td><td>Row 1 Column3</td></tr>
<tr><td>Row 2 Column2</td><td>Row 2 Column3</td></tr>
<tr><td colspan="3">Row 3 Column1</td></tr>

</table>

```

This will produce following result:

Column 1	Column 2	Column 3
Row 1 Column1	Row 1 Column2	Row 1 Column3
	Row 2 Column2	Row 2 Column3
Row 3 Column1		

3.2.6 The Width and Height Attributes

The Width Attribute. This attribute specifies the width of the table in pixel or in percentage of the Web page. The table will stretch to fit the entire width of the page if 100% is used. If width is not specified, the browser assumes the width of a particular table automatically by using the width of the element and text it contains. The essence of this attribute is to customize or control the display of tables on the Web page.

The height Attribute. This attribute specifies the height of the table in either pixel or percentage of the Web page. This is more commonly used on <tr> and <td> tags.

You can specify table width or height in terms of integer value or in terms of percentage of available screen area. Study example 3.1 for a better understanding of the use of these attributes

Example 3.1

```

<table border="1" width="300" height="120">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>

```



```
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
```

```
</table>
```

This will produce following result:

Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

4.0 CONCLUSION

One of the most important features of HTML is its support for tables. Tables help to present data in more understandable ways in a browser. HTML tables work in a fashion similar to tables in Microsoft Word in that it allows authors to build columns and rows containing most if the other HTML elements and contents.

5.0 SUMMARY

HTML allows you to create table by using table tags in conjunction with the tr and td as the basic tags. Important attributes which provide additional information about table elements were also covered in this unit.

6.0 TUTOR MARKED ASSIGNMENTS

How are tables more useful as layout mechanism than plain HTML? Give an example that show their usefulness in this regards

7.0 REFERENCES / FURTHER READINGS

Barbara K. Kaye, Norman J. Medoff (2001) The World Wide Web: A Mass Communication Perspective, Mayfield Publishing Company, USA

Barrie Sosinsky, Valda Hilley, (2004) Programming the Web: An Introduction, McGraw-Hill, USA

Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Deitel P. J., and Deitel H.M (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

June Cohen (2003), The Unusually Useful Web Book, New Rider, USA

Marc D. Miller, Thomas C. Padgett (2003) Web Design Using Macromedia Dreamweaver, McGraw-Hill/ Irwin, USA

Peter Weverka (2001) Instant Web Pages, Sybex Inc, USA

Raymond Greenlaw and Ellen Hepp (2001), In-Line/On-Line Fundamentals of the Internet and World Wide Web, McGraw-Hill Learning Center, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

http://www.quackit.com/html/html_table_tutorial.cfm

<http://www.w3.org/TR/html4/struct/tables.html#h-11.1>

UNIT 4: INPUT TAGS AND FORM PROCESSING

CONTENTS

- 1.0 INTRODUCTION
- 2.0 OBJECTIVES
- 3.0 MAIN CONTENT
 - 3.1 The Form Element
 - 3.2 HTML – Text Field /Box
 - 3.3 HTML – Password Field
 - 3.4 HTML – Checkbox Forms
 - 3.5 HTML –Radio Form
 - 3.6 HTML - Textarea
 - 3.7 HTML –Selection Forms and Drop Down Lists
 - 3.8 Submit Button
- 4.0 CONCLUSION
- 5.0 SUMMARY
- 6.0 TUTOR MARKED ASSIGNMENT
- 7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

A form is an HTML element that contains and organizes other objects or controls. Form elements are like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc. which are used to take information from the user. Generally user complete a form by entering text, selecting menu items, etc. before submitting the it to an agent such as Web server, a mail server, etc. for processing. Within the server, the back-end application such as CGI, ASP Script or PHP script etc does the required processing on the users data as specified in the Scripts are rendered to the Web browsers if there is a need to do so.

2.0 OBJECTIVES

At this end of this unit, the student should be able to:

- explain the meaning of Forms
- write HTML codes to process form information
- explain how to use Form action and Methods
- discuss and use various form elements

3.0 Main Contents

3.1 The Form Element

The <form> tag is used to specify the beginning of a form area on a Web page while the </form> tag is used to specify the end of a form area. Common attributes used to the <form> tags are

name, method, and action. These attributes are used to specify what server-side program or file will process the form, how the form information will be sent to the server, and the name of the form.

Name Attribute- This is an optional attribute and names the form. The name of the form is required for it to be easily access by client scripting languages, such as JavaScript (will be treated latter) to edit and verify the form prior to sending its information for server-side processing

Method Attribute

This attribute is optional. The values GET or POST may be specified. When GET is specified, it causes the form data to be appended to the URL and sent to the Web server. When the value POST is specified for the attribute, it transmits the form data in the body of the HTTP response. This is a more referred and acceptable method.

Example 4.1 is a sample HTML fragment that defines a simple form that allows the user to enter a first name, last name, department and the gender of a user. When the submit button is activated, the form will be processed based on the instructions contained a PHP script (process.php).

Action Attribute

This attribute is optional. However, when a value is specified it determines the server-side program or script that will process your form data using CGI. The value should be a valid file name on a Web server. Examples PHP script(.php extension), Sun JavaServer Pages(.jsp extension), Microsoft Active Server Pages(.asp extension).

Example 4.1: HTML code for Form

```
<FORM action="process.php" method="post">
  <P>
    First name: <INPUT type="text" name="firstname"><BR>
    Last name: <INPUT type="text" name="lastname"><BR>
    Department: <INPUT type="text" name="dept"><BR>
    <INPUT type="radio" name="gender" value="Male"> Male<BR>
    <INPUT type="radio" name="gender" value="Female"> Female<BR>
    <INPUT type="submit" value="Send"> <INPUT type="reset" value="Reset" >
  </P>
</FORM>
```

Output

First name:

Last name:

Department:

Male

Female

Figure 1: Form Display on Browser

3.2 HTML – Text Field /Box

Text fields are small rectangles that allow a user to simply input some text or numeric information, such as names, e-mail addresses, phone number, and other text and submit the information to the web server. The form element is configured by the `<input />` tag. Common attributes of a text box are `type`, `size`, `maxlength`, `value` and `password`.

HTML – Text Field Size

You can control the size of the text area by specifying the `size` attribute. The example below provides 3 different sizes for your text fields. The default size is usually around 20 characters long. See example 2

Example 2: HTML Code for Input / Text Box Element

```
<html>
<body>
<FORM action="process.php" method="post">
  <P>
    First name: <INPUT type="text" name="firstname" size="20"><BR>
    Last name: <INPUT type="text" name="lastname" size="19"><BR>
    Department: <INPUT type="text" name="dept" size="15"><BR>
  </P>
</FORM>
</body>
</html>
```

Output

First name:

Last name:

Department:

Figure 2: Textbox

HTML –TextField Maxlength

When the value is specified, it is used to limit the number of characters a user can type into fields, it is a good programming practice to specify the a maxlength, generally this should match the size of your field.

Example 3: HTML Code Textbox with Maxlength Attribute:

```
<html>
<body>
<FORM action="process.php" method="post">
  <P>
    First name: <INPUT type="text" name="firstname" size="20" maxlength="20"><BR>
    Last name: <INPUT type="text" name="lastname" size="20" maxlength="20"><BR>
    Department: <INPUT type="text" name="dept" size="15" maxlength="15"><BR>
  </P>
</FORM>
</body>
</html>
```

Practice 1

Run the program and attempt to enter data more than the maximum length specified. Write down your experience in a sentence?

HTML –TextField Value

The value attribute, is used to pre-populate your text fields with some information. This can then be manipulated with any scripting language such as PHP, PERL, etc. See example 4

Example 4 HTML Code for TextField value:

```
<html>
<body>
<FORM action="process.php" method="post">
  <P>
    First name: <INPUT type="text" name="firstname" size="20" maxlength="20"
value="Sule"><BR>
    Last name: <INPUT type="text" name="lastname" size="20" maxlength="20"
Value="Okeke"><BR>
    Department: <INPUT type="text" name="dept" size="15" maxlength="15"
Value="Economics"><BR>
```

```

    </P>
  </FORM>
</body>
</html>

```

First name:

Last name:

Department:

Figure 3: Text Field with Values:

3.3 HTML – Password Field

Password fields are a special type of `<input />` tag. To implement them change the type attribute from `text` to `password`. It is used to accept information that need to be hidden as it is entered. When a user types in information in a password box, asterisks (i.e `*`) are displayed instead of the characters that are being typed. This does not mean that the data entered is encrypted. To encrypt you data you must use a scripting language to process the data captured. See example 5

Example 5: Password Attribute

HTML Code:

```

<html>
<body>
<FORM action="process.php" method="post">
  <P>
    MatNo: <INPUT type="text" name="matno" size="11" maxlength="11"><BR>
    User Name: <INPUT type="text" name="username" size="20" maxlength="20"><BR>
    Password: <INPUT type="password" name="pass" size="8" maxlength="8"><BR>
  </P>
</FORM>
</body>
</html>

```

MatNo:

User Name:

Password:

Figure 4: Password Fields

3.4 HTML – Checkbox Forms

Checkboxes are another type of `<input />` form and are used for instances where a user may wish to select some or all multiple options. The type attribute must be set to checkbox and you also need to set the name and value attributes. A sample checkbox code and the corresponding form are shown in Example 6 and Figure 5

Example 6: HTML Code for Checkbox:

```
<html>
<body>
<FORM action="process.php" method="post">
<P>
<p>Please select the courses you want to register for the semester.</p>
CIT313: <input type="checkbox" name="courses" value="CIT313" /><br />
CIT314: <input type="checkbox" name="courses" value="CIT314" /><br />
CIT315: <input type="checkbox" name="courses" value="CIT315" /><br />
EDU325: <input type="checkbox" name="courses" value="EDU325" /><br />
</P>
</FORM>
</body>
</html>
```

HTML –Checkboxes selected

With checkboxes, you could pre check the input boxes for your viewers using the checked attribute. For example if the course CIT313 is a compulsory course for all the students for a semester, it can be pre-checked. To implement this, simply set the checked attribute to yes. The codes for implementing checkboxes selected and the corresponding output are shown in example 7 and Figure 5 respectively.

Example 7: HTML Cod for Checkbox selected:

```
<p>Please select the courses you want to register for the semester.</p >
CIT313: <input type="checkbox" checked = yes name="courses" value="CIT313" /><br />
CIT314: <input type="checkbox" name="courses" value="CIT314" /><br />
CIT315: <input type="checkbox" name="courses" value="CIT315" /><br />
EDU325: <input type="checkbox" name="courses" value="EDU325" /><br />
</P>
```

Output

Please select the courses you want to register for the semester.

CIT313:
 CIT314:
 CIT315:
 EDU325:

Figure 5 Checkbox selected

3.5 HTML –Radio Form

Radios are types of input forms that allow a user to select exactly one item from a group of predetermined items. In order to achieve this, we must properly name each radio button selection accordingly. Thus each radio button in a group is given the same name and a unique value. The codes for implementing radio buttons and the corresponding output are shown in example 8 and Figure 6 respectively.

Example 9 HTML Code for Radio Button:

Economics: `<input type="radio" name="department" />`
 Computer Science: `<input type="radio" name="department" />`
 Accounting: `<input type="radio" name="department" />`

Economics:
 Computer Science:
 Accounting:

Figure 6 Radio Button

By naming these three radios "department" they are identified as being related by the browser and.

HTML –Radio Checked

By using the checked attribute, are able to configure the radio button to be selected by default when displayed by the browser. Example 10 shows the codes for Radio Checked while Figure 7 shows the corresponding output on a browser.

Example 10: HTML Code for Radio Checked:

Economics: `<input type="radio" name="department" checked="yes" />`

Computer Science: `<input type="radio" name="department" />`

Accounting: `<input type="radio" name="department" />`

Economics:

Computer Science:

Accounting:

Figure 7: Radio Checked

3.6 HTML - Textarea

This allows multi text field to be entered by a user. Paragraphs, essays, questions, descriptions or memos can be cut and pasted into textareas and submitted. Textareas have an opening tag `<textarea>` and a closing tag `</textarea>`. Example 11 and Figure 8 show the codes and the output of textarea on a browser.

Example 3: HTML Code for Textarea:

`<textarea>National Open University of Nigeria!</textarea>`

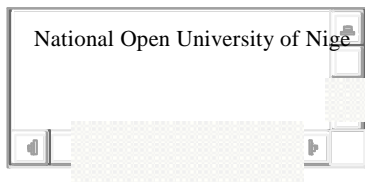


Figure 8: Textarea

HTML – Text area Col and Rows

To adjust the size of the appearance of the text area requires two attributes, cols and rows with numeric values. The larger the value the larger the field will appear. Example 12 is a sample code for implementing text area col and row.

Example 12: HTML Code for col and row in textarea

`<textarea cols="30" rows="10">Text Area!</textarea>`

`<textarea cols="10" rows="2">Text Area!</textarea>`

`<textarea cols="25" rows="5">Text Area!</textarea>`

3.7 HTML –Selection Forms and Drop Down Lists

Drop down lists are the basic selection forms. The `<select>` container tag along with the `<option>` tags are used to configure the select list. Other names used to describe the drop down list are select list, select box, drop-down box, and option box. Drop down lists have options a user can select. A sample code to implement Drop Down list and the corresponding output on a browser are shown in example 13 and Figure 9.

Example 13: HTML Code for Drop Down List:

```
<select>
<option>Economic </option>
<option>Computer Science </option>
<option>Accounting </option>
</select>
```

By default the first coded `<option>` will be displayed or selected as the default. We can change this using the `selected` attribute.

Example 14: HTML Code for Drop Down list with selected attribute:

```
<select>
<option>Economic </option>
<option>Computer Science </option>
<option selected= yes >Accounting </option>
</select>
```

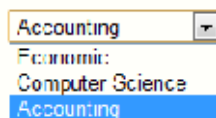


Figure 10: Drop Down list

3.8 Submit Button

Submit

Submission buttons are a type of `<input />` tag and is used to submit the form. To achieve form submission set the type attribute to `submit`. This creates a special type of button in your forms that will cause the browser to send the form data to the web server provided there are available server scripting codes to achieve this.

HTML – Reset Button

Reset

Reset buttons exist to reset the fields of your form to its initial values. See the codes in example 14.

Example 14: HTML Code for Submit and Reset Buttons

```
<input type="submit" value="Submit" /><br />  
<input type="Reset" value="Reset" /><br />
```

Notice that in the above example we also changed what was written on our button using the *value attribute*. This can be changed to any value you wish.

Practice 2

Review the code in example 4.1, make some changes to it and run it

4.0 CONCLUSION

Forms are important because they help Web page visitors to view data and gather or submit information to a server. Form elements include text boxes, labels, lists, check boxes and radio buttons. HTML tags that configure these form elements include the <input/> and <textarea> tag. Forms usually need to call some type of server-side processing scripts to perform functions such as updating a database, submitting form data and sending a mail. The server side script can be written in PHP (Hypertext Pre-processor) language, JSP(Java Server Pages), ColdFusion, etc

5.0 SUMMARY

In this unit, we covered how to create and use form on Web Pages. You also learnt how to configure elements of form such as text boxes, labels, checkboxes etc. In the next module we shall be looking at how to design more attractive and interactive Web pages.

6.0 TUTOR MARKED ASSIGNMENT

1.)Which attribute of the <form> tag is used to specify the name and the location of the script that will process the form field values

a) action b) process c) method d) none of the above

2)You would like to conduct a survey and ask you Web page visitors to vote for their favourite candidate in an election. Which of the following form elements is best to use for this purpose?

a) check box b) radio button c) text box d) scrolling text box

7.0 REFERENCES / FURTHER READINGS

Barbara K. Kaye, Norman J. Medoff (2001) The World Wide Web: A Mass Communication Perspective, Mayfield Publishing Company, USA

Barrie Sosinsky, Valda Hilley, (2004) Programming the Web: An Introduction, McGraw-Hill, USA

Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Deitel P. J., and Deitel H.M (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

June Cohen (2003), The Unusually Useful Web Book, New Rider, USA

Marc D. Miller, Thomas C. Padgett (2003) Web Design Using Macromedia Dreamweaver, McGraw-Hill/ Irwin, USA

Peter Weverka (2001) Instant Web Pages, Sybex Inc, USA

Raymond Greenlaw and Ellen Hepp (2001), In-Line/On-Line Fundamentals of the Internet and World Wide Web, McGraw-Hill Learning Center, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

MODULE THREE

UNIT 1: CASCADING STYLE SHEET CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES

3.0 MAIN CONTENT

3.1 Advantages of Cascading Style Sheets.

3.2 Syntax of CSS

3.3 CSS Comments

3.4 Types of Cascading Style Sheets

3.5 Using CSS Classes

3.6 Understanding CSS Background

3.7 Using CSS background image

3.8 CSS Fonts

4.0 CONCLUSION

5.0 SUMMARY

6.0 TUTOR MARKED ASSIGNMENT

7.0 REFERENCES / FURTHER READINGS

1.0 Introduction

CSS stands for Cascading Style Sheets. It defines how to display HTML elements. HTML was never intended to contain tags for formatting a document. It came as after thoughts. HTML was intended to define the content of a document, like:

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

The use CSS enables all formatting to be removed from HTML document, and stored in separate CSS file. CSS helps to control the look and feel of HTML documents in an organized and efficient manner. Thus with the use of CSS a designer will be able to achieve the following:

- Add new looks to HTML pages

- Completely restyle a web site with only a few changes to your CSS code

- Use the "style" you create on any webpage you wish!

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to explain the meaning and the importance of CSS

- to use CSS format web pages
- to add CSS to HTML files

3.0 Main Content

3.1 Advantages of Cascading Style Sheets.

Typography and page layout can be better controlled. These features include font size, line spacing, letter spacing, indent, margin, and element positioning

Style is separable from structure. The format of the text and colors used on the page can be configured and stored separately from the body section of the Web page document

Documents are potentially smaller: The formatting is separate from the documents, therefore, the actual document should be smaller

Style can be stored. This makes site maintenance easier. Again, if the style needs to be changed, it's possible to complete the modifications by changing the style sheet only.

3.2 Syntax of CSS

Style sheets are made up of rules that describe the styling to be applied. A CSS rule has two main parts: a selector, and one or more declarations:

Selector	Declaration	Declaration
h1	{ Color: green ; Property Value	text-align:center;} Property Value

In the above format, the selector is the heart and soul of CSS. It defines which HTML elements you are going to be manipulating with CSS code. Each declaration consists of a property and a value. The property is the style attribute you want to change and each property has a value. A CSS declaration always ends with a semicolon, while declaration groups are surrounded by curly brackets:

Example 1a: CSS Code

```
p {color:green;text-align:center;}
```

The codes in Example 1a can also be written as in Example 1b to make the CSS more readable by putting one declaration on each line, as follows:

Example 1b: CSS Code

```
p
{
color:green;
text-align:center;
}
```

3.3 CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are not executed or run by browsers. Comments in CSS is demonstrated in example 2

A CSS comment begins with "/*", and ends with "*/", like this:

```
/*This is a comment*/
```

Example 2: Comments in CSS

```
p
{
text-align:center;
/*This is another comment*/
color:black;
font-family:arial;
}
```

3.4 Types of Cascading Style Sheets

Cascading Style Sheets come in mainly in three flavors: internal, external, and inline. We shall be studying these flavours in this unit.

3.4.1 Internal CSS

This is defined in the header of a Web page. And apply to the Web page document. To use it, you must add a new tag, <style>, inside the <head> tag. The HTML code below contains an example of internal style <style>'s usage.

Example 3: Internal CSS code

```
<html>
<head>
<style type="text/css">
</style>
</head>
<body>
<p> National Open University of Nigeria </p>
</body>
</html>
```


The effect of the use of CSS is not easily noticed in this code. The code style tag just tells the browser that we will be defining some CSS to be used on this page.

Creating Internal CSS code

CSS codes are written to manipulate existing HTML codes. Example 4 represent the use of Internal CSS codes in a simple and function Web page design.

Example 4: Internal CSS Code:

```
<html>
<head>
<style type="text/css">
p {color: white; }
body {background-color: black; }
</style>
</head>
<body>
<p>This displays National Open University using white text on a black background!</p>
</body>
</html>
```

Display:

This displays National Open University using white texts on a black background!

Figure 1: Internal CSS

What we did in the above example was to manipulate <p> and <body> which are common HTML tags. In addition we went through the following steps:

Stage 1

We chose the HTML element we wanted to manipulate. - p { : ; }
 Then we chose the CSS attribute color. - p { color: ; }
 Next we choose the font color to be white. - p { color: white; }

Stage 2

We choose the HTML element Body - body { : ; }
 Then we chose the CSS attribute. - body { background-color: ; }
 Next we chose the background color to be black. - body { background-color:black; }

3.4.2 Using External CSS

The use of external CSS helps to keep CSS separate from HTML codes. Placing CSS in a separate file allows for easier maintenance of web pages. To use external CSS file, you need to first create the required CSS code and save it with a ".css" file extension. This CSS file is then referenced in your HTML using the <link> instead of <style> tag.

Example 5: Creating and using external CSS

```
h3{ color: red; }
p {color: white; }
body {background-color: black; }
```

Practice 1:

Open up notepad.exe, or any other plain text editor and type the CSS codes in Example 5 above. Now save the file as a CSS (.css) file. Make sure that you are not saving it as a text (.txt) file, as notepad likes to do by default. Name the file "test.css" (without the quotes). Now create a new HTML file and fill it with the following code. Then save this file as "example6.html" (without the quotes) in the same directory as your CSS file. Now open your HTML file in your browser and compare with what we have in Figure 2

Example 6: HTML Code:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="test.css" />
</head>
<body>
<h3> Displays WELCOME with the style (red color) specified for h3 </h3>
<p>This displays National Open University using white text on a black background!</p>
</body>
</html>
```

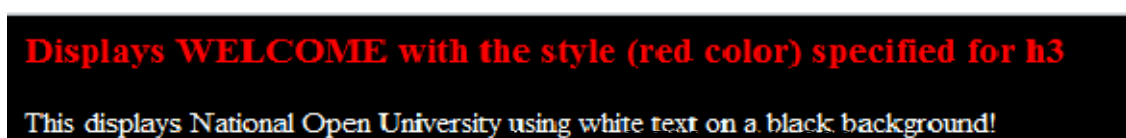


Figure 2. HTML with CSS

3.4.3 Using CSS Inline

Inline style can be coded in the body of the Web pages as an attribute of HTML tag. Inline CSS has the highest priority out of external, internal, and inline CSS. This means the inline style can override styles that are defined in external or internal. To add a style inside an HTML element all

you need to do is specify the desired CSS properties with the style HTML attribute. Study Example 7 and Figure 3.

Example 7: CSS Inline Code:

```
<p style="background: blue; color: white;"> This displays National Open University with white
color on a blue background
</p>
```

This displays National Open University of Nigeria with white color on a blue background

Figure 3: Inline CSS

3.5 Using CSS Classes

CSS classes allow you to apply a CSS rule to certain class of elements on a Web page and not necessarily tie the style to a particular HTML tag. It also allows you to give an HTML element multiple looks with CSS. Say for example that sometimes you want the font to be large and white, while other times you would prefer the font to be small and black for a particular HTML items e.g Paragraph <p>.

Using classes is simple. You just need to add an extension to the typical CSS code and make sure you specify this extension in your HTML. Let's try this with an example of making two paragraphs that behave differently. First we begin with the CSS, note the red text.

```
p.first{ color: blue; }
p.second{ color: red; }
```

Example 9: HTML Code using CSS Classes:

```
<html>
<body>
<p>This displays National Open University of Nigeria in a normal paragraph.</p>
<p class="first">This displays National Open University of Nigeria in a paragraph that uses the
p.first CSS code!</p>
<p class="second"> This displays National Open University of Nigeria in a paragraph that uses
the p.second CSS code! </p>
...
```

This displays National Open University of Nigeria in a normal paragraph.
--

This displays National Open University of Nigeria in a paragraph that uses the p.first CSS code!
--

This displays National Open University of Nigeria in a paragraph that uses the p.sceond CSS code!

Figure 4: Display:

3.6 Understanding CSS Background

The background of your website is very important. With CSS, you are able to set the background color of image of any CSS element. With CSS, a color is most often specified by:

a HEX value - like "#ff0000"
 an RGB value - like "rgb(255,0,0)"
 a color name - like "red"

Colors can be produced for a number of page elements using the color names or RGB, hexadecimal codes as demonstrated in example

Example 10.CSS Background Code:

```
h4 { background-color: white; }
p { background-color: #1078E1; }
body { background-color: rgb( 149, 206, 145); }
```

3.7 Using CSS background image

The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element. The background image for a page can be set as demonstrated in Example 11.

Example 11: CSS Background image Code:

```
p { background-image: url(myPic.jpg); }
body { background-image:url('logo.gif');}
h4{ background-image: url(http://www.nou.ed.ng /pics/cssPic/logo.gif); }
```

3.8 CSS Fonts

CSS gives you great control over the way your text is displayed. You can change the text size, color, style, and lots more.

3.8.1 CSS font color

CSS colors are defined using a hexadecimal (hex) notation for the combination of Red, Green, and Blue color values (RGB). The lowest value that can be given to one of the light sources is 0 (hex 00). The highest value is 255 (hex FF). This is illustrated with Example 12.

Example 12: CSS Font Color Code

```
h4 {color: red;}
h5 {color: #9000A1;}
h6 {color: rgb (0,220, 60);}
```

3.8.2 CSS font family

Font families may be assigned by a specific font name or a generic font family. This is illustrated by Example 13.

Example 13: CSS Font Family Code:

```
h4 { font-family: sans-serif; }
h5 { font-family: serif; }
h6 { font-family: arial; }
```

3.8.3 CSS Font Size

Font size are used to manipulate the size of your fonts by using values, percentages, or ~~key~~ terms. Key terms are not very useful, the common terms are: xx-large, x-large, large, medium, small, x-small, and xx-small. You use values if you do not want the user to be able to increase the size of the font. Percentages are used when you want to change the default font, but do not want to set a static value. This is illustrated by Example 14.

Example 14: CSS Font Size Code:

```
p { font-size: 120%; }
body { font-size: 10px; }
h1 { font-size: x-large; }
```

3.8.4 Text Align

By default, texts on any website are aligned to the left. If you require different alignment, you can specify it using the text-align attribute. This is illustrated in Example 15.

Example 15: CSS Text Align Code:

```
p { text-align: right; }
h2 { text-align: justify; }
```

3.8.5 Text Transform

Text-transform is used to modify the capitalization of your text. This is illustrated in Example 16.

Example 16: CSS Text Transform Code:

```
p { text-transform: capitalize; }
h5 { text-transform: uppercase; }
h6 { text-transform: lowercase; }
```

4.0 CONCLUSION

Cascading style sheets refers to a unique processing approach by which multiple style sheets may be applied to single Web page but is supported by all style sheet languages. There are lots of advantages which results from using style sheets to apply stylistic formatting to Web pages. For example, a Web developer can separate documents available to many pages in a Web site, thus enabling the developer to apply similar styles across many documents while having to modify styles in one file when there is a need.

5.0 SUMMARY

Cascading style sheets were discussed in this unit as means of providing a uniform and consistent presentation detail in HTML documents. The three levels i.e Internal, External and Inline in which style sheet can appear were covered. CSS classes were also covered.

6.0 TUTOR MARKED ASSIGNMENT

- a) What is a style sheet selector?
- b) How may styles be referenced in an HTML page?
- c) What property can be set in order to change the color of paragraph text to red (from the default black)?
- d) Write an internal style for producing an <H2> heading tag that has a blue background, is in italics, and is centred.

7.0 REFERENCES / FURTHER READINGS

Barbara K. Kaye, Norman J. Medoff (2001) *The World Wide Web: A Mass Communication Perspective*, Mayfield Publishing Company, USA

Barrie Sosinsky, Valda Hilley, (2004) *Programming the Web: An Introduction*, McGraw-Hill, USA

Behrouz A Forouzan (2003), *Data Communications and Networking*, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Deitel P. J., and Deitel H.M (2008), *Internet and World Wide Web: How to Program*, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

June Cohen (2003), *The Unusually Useful Web Book*, New Rider, USA

Marc D. Miller, Thomas C. Padgett (2003) *Web Design Using Macromedia Dreamweaver*, McGraw-Hill/ Irwin, USA

Peter Weverka (2001) Instant Web Pages, Sybex Inc, USA

Raymond Greenlaw and Ellen Hepp (2001), In-Line/On-Line Fundamentals of the Internet and World Wide Web, McGraw-Hill Learning Center, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

<http://www.tizag.com/cssT/selector.php>

http://www.w3schools.com/css/css_syntax.asp

UNIT 2: FUNDAMENTALS OF JAVASCRIPT

CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES

3.0 MAIN CONTENT

3.1 Inserting a JavaScript into an HTML page

3.2 JavaScript Comments

3.3 Where to Locate JavaScript in a Program

3.4 JavaScript Statements

3.5 JavaScript Variables

3.6 Arithmetic Operators and Expressions

4.0 CONCLUSION

5.0 SUMMARY

6.0 TUTOR MARKED ASSIGNMENT

7.0 REFERENCES / FURTHER READINGS

1.0 Introduction

JavaScript is a scripting language that was developed by Netscape Communicator to provide interactivity to static Web pages. The language was originally developed by Netscape under the name LiveScript in late 1995. Netscape and Sun in December, 1995 later released LiveScript under the name JavaScript. Many people seem to be confused about the relationship of JavaScript and Java which is a separate programming language. JavaScript is a simple, interpreted language while Java whereas Java is a compiled object-oriented programming language. In this unit, we shall describe basic facts about JavaScript and how to incorporate the most commonly used JavaScript elements into pages.

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to explain the meaning of JavaScript
- to write simple JavaScript programs
- to use arithmetic operators
- to use JavaScript to make web pages interactive

3.0 Main Contents

3.1 Inserting a JavaScript into an HTML page

To insert a JavaScript into an HTML page, we use the `<script>` tag. Inside the `<script>` tag we use the `type` attribute to define the scripting language.

So, we have `<script type="text/javascript">` and `</script>` to connote where the JavaScript starts and ends: Example 1 is a simple JavaScript code that displays on a browser Welcome to National Open University of Nigeria – without the quotes.

Example 1: Simple JavaScript Code

```
<html>
<body>
<script type= text/javascript >
document.write( Welcome to National Open University of Nigeria );
</script>
</body>
</html>
```

In this example, the `document.write` command is a standard JavaScript command for writing output to a page. By entering the `document.write` command between the `<script>` and `</script>` tags, the browser will recognize it as a JavaScript command and execute the code line.

3.2 JavaScript Comments

Comments are added to JavaScripts codes to make them more readable. JavaScript allows the use of single line or multiple lines comments. To put a comment on a single line use `//`. Example 2 illustrates the use of a comment in JavaScript codes.

Example 2: Comments in JavaScript

```
<script type="text/javascript">
// My details are as d isplayed in the following three paragraphs
document.write("<p>My Name is Ade Musa Okeke </p>");
document.write("<p>I am in the School of Science and Technology.</p>");
document.write("<p>My Matriculation Number is NOU0311 11.</p>");
</script>
```

To use multi line comments start with `/*` and end with `*/`. Example 4 is used to illustrate the use of multi line comments (`/* */`).

Example 3: Multi line comments

```
<script type="text/javascript">
/*
My details as a student of the National Open University of Nigeria
are displayed in the next three paragraphs
*/
document.write("<p>My Name is Ade Musa Okeke </p>");
document.write("<p>I am in the School of Science and Technology.</p>");
document.write("<p>My Matriculation Number is NOU0311 11.</p>");
</script>
```

3.3 Where to Locate JavaScript in a Program

JavaScript code can be located internally within the program or externally. If it is to be within the program, then it has to be located in the body or head section of an HTML page. Program instructions are executed sequentially, scripts that are to be executed later or when a user clicks a button are better placed in a function. For easy maintenance of programs it is better to separate your function from the main page content by locating them in the head section.

Example 4: JavaScript Codes located in the head section

```
<html>
<head>
<script type="text/javascript">
function message()
{
alert("This alert box was called with the onload event");
}
</script>
</head>

<body onload="message()">
</body>
</html>
```

If you don't want your script to be placed inside a function, or if your script should write page content, it should be placed in the body section.

Example 5: JavaScript Codes located in the body section

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
document.write("This message is written by JavaScript");
</script>
</body>
</html>
```

3.3.1 Using an External JavaScript

To use JavaScript as external file first it has to be written and saved with a .js file extension. Then point to the .js file in the "src" attribute of the <script> tag: Example 4 illustrates the use of JavaScript as an external file.

Example 4: External JavaScript

```
<html>
<head>
<script type="text/javascript" src="extfile.js"></script>
```

```

</head>
<body>
</body>
</html>

```

3.4 JavaScript Statements

JavaScript is a sequence of statements to be executed by the browser. Each statement must be separated by a semi colon. Example 5 is used to illustrate a programme that displays the details of student to the Web page

Example 5: Student details

```

<script type="text/javascript">
document.write("<p>My Name is Ade Musa Okeke </p>");
document.write("<p>I am in the School of Science and Technology.</p>");
document.write("<p>My Matriculation Number is NOU031111.</p>");
</script>

```

3.4.1 JavaScript Blocks

JavaScript statements can be grouped together in blocks. Blocks start with a left curly bracket {, and ends with a right curly bracket }. The purpose of a block is to make the sequence of statements execute together. Example 6 the three lines of the student's details are treated as a block.

Example 6: Block Statements

```

<script type="text/javascript">
{
document.write("<p>My Name is Ade Musa Okeke </p>");
document.write("<p>I am in the School of Science and Technology.</p>");
document.write("<p>My Matriculation Number is NOU031111.</p>");
}
</script>

```

3.5 JavaScript Variables

Variables are "containers" for storing information. As with algebra, JavaScript variables are used to hold values or expressions. A variable can have a short name, like amt, or a descriptive name, like amount

Rules for JavaScript variable names:

Variable names are case sensitive (the variable amt and AMT are two different variables)

Variable names must begin with a letter or the underscore character

Declaring (Creating) JavaScript Variables

A variable is declared by preceding it with the keyword `var`. Example 7 shows valid declaration of variables in JavaScript.

Example 7: Declaration Statements

```
var x;  
var myname;  
var examscore  
var radius  
var greetings;
```

Assignment Statement

As long as no values are assigned to variable, they will remain empty. To assign values to the variables use the assignment operator (`=`). You will learn about other operators later in this module. In Example 8, we combined both the declaration and assignment statements.

Example 8: Assignment and declaration Statement

```
var x=5;  
var myname= Adebola ;  
var examscore=89;  
var radius=1.0;  
var greetings= Welcome ;
```

In Example 8, variable `x` holds the value 5, `myname` holds the value Adebola, `examscore` holds the value 89, `radius` holds the value 1.0 while `greetings` holds the value Welcome. Note the use of quotes in the assignment of a text value to variables `myname` and `greetings` and the use of semicolon after each variable declaration. Semicolon is use JavaScript to mark the end of statement

JavaScript also allows you to assign a value to variable that has not been declared. See the examples below:

```
amt =10;
```

This is the same as

```
var Amt =10;
```

3.6 Arithmetic Operators and Expressions

An arithmetic expression is one which is evaluated by performing a sequence of arithmetic operations to obtain a numeric value to replace the expression. Arithmetic operators are used to perform arithmetic between variables and/or values. Table 1 shows a list of arithmetic operator and expressions.

.Given that Y=10, the table below explains the arithmetic operators:

Operators	Meaning	Example	Result
+	Addition	X=Y+2	X=12
-	Subtraction	X=Y-2	X=8
*	Multiplication	X=Y*2	X=20
/	Division	X=Y/2	X=5
%	Modulus	X=Y%2	X=0
++	Increment	X++	X=11
--	Decrement	X--	X=9

Table 1: Arithmetic operators and expressions

The list above is similar to that of basic mathematics. The only symbol that might look new to you is the modulus ("%"), which divides one operand by another and returns the remainder as its result. In addition, the + operator can be used to add string variables or text values together.

To add two or more string variables together, use the + operator.

```
txt1="National Open";
txt2="University of Nigeria";
txt3=txt1+txt2;
```

After the execution of the statements above, the variable txt3 contains "National Open University of Nigeria".

Practice 1

The program below computers the area of a circle. Type the codes using a text editor preferably notepad.exe. Save an HTML file and open it with a browser. What is the result?

```
<html>
<body>
<script type="text/javascript">
var radius=5;
var area =radius * radius * 3.14159
document.write("The Area of the Circle with radius=5 +area);
document.write("<br />");
}
</script>
```

```
</body>  
</html>
```

4.0 CONCLUSION

JavaScript statements are typically embedded directly with HTML. A single HTML document can include any number of embedded scripts. When used properly, JavaScript have the capacity in improve the look and enhance users interactivity with Web pages. Some statements that will enable you to write simple JavaScript code have been covered in this unit.

5.0 SUMMARY

JavaScript is the most popular scripting language of the Internet. It is majorly used as a client-side scripting language to add inter active functionality, validate forms, detect browsers, etc. in Web design. Some of its constructs have been covered in this unit. It is supported by major browsers, such as Internet Explorer, Fir efox, Chrome, Oper a, and Safari.

6.0 TUTOR MARKED ASSIGNMENTS

- a) Who developed JavaScript and when? Which browser support JavaScript?
- b) Locate a JavaScript calculator and explain how it works?
- c) Using JavaScript design a Web page that conver ts temperature reading in Celsius to Fahrenheit scale.

7.0 REFERENCES / FURTHER READINGS

Barbara K. Kaye, Norman J. Medoff (2001) The World Wide Web: A Mass Communication Perspective, Mayfield Publishing Company, USA

Barrie Sosinsky, Valda Hilley, (2004) Programming the Web: An Introduction, McGraw-Hill, USA

Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Dave Mercer, Schaum's Outline of HTML, Introduction to Web Page Design and Development McGraw-Hill, USA, 2002

Deitel P. J., and Deitel H.M (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

June Cohen (2003), The Unusually Useful Web Book, New Rider, USA

Marc D. Miller, Thomas C. Padgett (2003) Web Design Using Macromedia Dreamweaver, McGraw-Hill/ Irwin, USA

Peter Weverka (2001) Instant Web Pages, Sybex Inc, USA

Raymond Greenlaw and Ellen Hepp (2001), In-Line/On-Line Fundamentals of the Internet and World Wide Web, McGraw-Hill Learning Center, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

UNIT 3: USING CONTROL STATEMENTS IN JAVASCRIPT

CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES

3.0 MAIN CONTENT

3.1 Logical Statement

3.2 Decision Making

3.3 Iteration on JavaScript

4.0 CONCLUSION

5.0 SUMMARY

6.0 TUTOR MARKED ASSIGNMENT

7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

JavaScript programs will be executed in the order in which statements are written except for the use of control statements with the scripts. The use of control statements can lead to the conditional, repeated and alteration of the normal sequential flow of control. Control statements in JavaScript are similar to their counterparts in C/C++ and Java. They are thus easy to learn.

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to implement logical construct with JavaScript
- to implement decision statements with JavaScript
- to implements loops with JavaScript

3.0 Main Content

3.1 Logical Statement

As you write your program you may want the some sets of statements to be exexecuted based on the outcome of your logical. Comparson and logical operators would be required. As the names connote, they allow you to compare values. They are used with if, while, switch, and for statements to accomlish decision or iterative constructs in programming. You will be interested in testing if one operand is greater than, less than, equal to, or not equal to another operand. The majority of these operators will probably look familiar as in other programming languages. The result you obtain is usually a true or false which which further determines which statement the computer should execute. You will require

Comparison Operators

Comparison operators are used in logical statements to determine equality or difference between variables or values. Given that Y=10, Table 2 explains the comparison operators:

Operators	Meaning	Example	Result
==	Equal to	Y == 8	False
===	Equivalent to	Y === 10	True
		Y === "10"	False
!=	Not Equal to	Y != 8	True
>	Greater than	Y > 8	True
<	Less than	Y < 8	False
>=	Greater or Equal to	Y >= 8	True
<=	Less or Equal to	Y <= 8	False

Logical Operators

Logical operators are used to determine the logic between variables or values. Given that X=5 and Y=10, the Table 3 explains the results of the use of logical operator in the expressions

Operators	Meaning	Example	Result
&&	And	(x < 7 && y > 6)	False
	Or	(x == 5 y == 6)	True
!	Not	!(x == 7)	True

Table 3: Logical Operators

3.2 Decision Making

You may wish to test the value of a variable, and perform different tasks based on the outcome of the test. For instance, you may need to check the examination score of a student to know whether he passed or failed and what grade he made. You can use conditional statements in your code to achieve this. Conditional statements are used to perform different actions based on different conditions. The if and switch commands are commonly used to implement the conditional statement. We shall briefly examine the different construct of the if and switch statements.

If Statement - this is used to execute some code only if a specified condition is true

Syntax

```
if (condition)
{
    code to be executed if condition is true
}
```

Example 1

```
<script type="text/javascript">
```

```
var examscore = 80;
    var result;
    if (examscore >= 70)
    {
        result = Pass ;
        document.write("<b>Congratulation, You Passed </b>");
    }
</script>
```

If...else Statement

This is used to execute some codes if the condition is true and another code if the condition is false

Syntax

```
if (condition)
{
    code to be executed if condition is true
}
else
{
    code to be executed if condition is not true
}
```

Example 2

```
<script type="text/javascript">
    var examscore = 80;
    var result;
    if (examscore >= 45)
    {
        result = Pass ;
        document.write("<b>Congratulation, You Passed </b>");
    }
    else

    {
        result = Fail ;
        document.write("<b> You Failed, Try again</b>");
    }
</script>
```

This will display the information Congratulation, You passed.

Practice 1

If the value of exam score is 35 what message will be displayed on the web browser?

Switch Statement

This is used to select one of many blocks of code to be executed. The syntax of the statement is:

Syntax

```
switch(m)
{
case 1:
    execute code block 1
    break;
case 2:
    execute code block 2
    break;
.
.
.
case m:
    execute code block m
    break;

default:
    code to be executed if m is different from case 1,Case 2, ...Case m
}
```

It works by evaluation a single expression *m* (most often a variable). The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. The `break` command is used to prevent the code from running into the next case automatically. We examine this by looking at a program that displays the day of the week based on a user selection

Example 3

```
<script type="text/javascript">
    var dayoftheWeek;
    switch (dayoftheWeek)
    {
        Case 1: document.write("<b>Today is Sunday </b>");
```

```
        break;
    }
    Case 2: document.write("<b>Today is Monday </b>");
    break;
    {
    Case 3: document.write("<b>Today is Tuesday </b>");
    break;
    }

    {
    Case 4: document.write("<b>Today is Wednesday</b>");
    break;
    }

    {
    Case 5: document.write("<b>Today is Thursday </b>");
    break;
    }
    {
    Case 6: document.write("<b>Today is Friday</b>");
    break;
    }
    {
    Case 7: document.write("<b>Today is Saturday </b>");
    break;
    }
    {
    Default: document.write("<b>There are 7 Days in a week </b>");
    break;
    }
</script>
```

3.3 Iteration on JavaScript

Some statements in JavaScript are known as iterative statements. Instead of adding several almost equal lines in a script we can use loops to perform the tasks. Loop statements have control structures that delimit them and which determine how many times (zero or more) the delimited code is executed, based on some condition.

We will look at two structures here

- o the for statement
- o the while statement and its variants

The for Loop

The syntax of the for statement is

```
for (startvalue ; condition; increment ) {  
statements;  
}
```

Notice that there are three variable inside the for statement conditional expression. There are

startvalue

This holds the value of the initial state of the variable to be tested. It is usually done as an assignment.

condition

The condition to be tested for. The statement keeps processing as long as it remains true.

increment

The increment by which the variable being tested changes.

Example 4

```
<html>  
<body>  
<script type="text/javascript">  
var num=0;  
for (i=0;num<=100;num+)  
{  
document.write("The Next No is " + num);  
document.write("<br />");  
}  
</script>  
</body>  
</html>
```

Example 4 defines a loop that starts with $i=0$. The loop will continue to run as long as i is less than, or equal to 100. i will increase by 1 each time the loop runs. The loop will generate integer numbers from 0 to 100 integer numbers.

The while statement

The while statement test a condition, and when true, repeatedly runs a block of code until the condition is no longer true.

The syntax is given as follows:

```
While (expression) {
```

```
Statements;  
}
```

Another way to accomplish the task in example 4 is by using a while loop statement as shown in Example 5. The loop starts with $i=0$. The loop will continue to run as long as i is less than, or equal to 100. i will increase by 1 each time the loop runs:

Example 5

```
<html>  
<body>  
<script type="text/javascript">  
var num=0;  
while (num<=100)  
{  
    document.write("The Next number is " +num);  
    document.write("<br />");  
}  
</script>  
</body>  
</html>
```

Example 6

The do... while statement

This is required when a block of code is to be run at least once. After running a block of code once, it evaluates the conditional expression. If the conditional expression is true, then it loops back to the beginning of the statement and starts again other it stops.

The syntax is as follow:

```
do {  
statements;  
}  
While (expression);
```

Example 7

```
<html>  
<body>  
<script type="text/javascript">  
var num=0;  
do  
{  
    document.write("The next number is " + num);
```

```
    document.write("<br />");
    }
while (num<=10);
</script>
</body>
</html>
```

4.0 CONCLUSION

The normal execution of statements in a program is one after the other in the order in which they are written. This process is called sequential execution. Programmer can however, specify the order in which statements should be executed by using control constructs/statement. Some of these constructs have been covered in this unit.

5.0 SUMMARY

In this unit, we have covered the basic statements required to implement Control Constructs in JavaScript. In this next we shall cover events and events handlers.

6.0 TUTOR MARKED ASSIGNMENT

a) Identify and correct the errors in following segments of code

```
if ( age >= 30 );
    document.write( Age greater than or equal to 30);
else
    document.write( Age is less than 30);
```

b) Write a script that outputs HTML text that keeps displaying in the browser window the multiples of the integer 2, namely 2,4, 8, 16, 32, 64, 128, etc. Ensure that your loop terminates when the value 2048576 is printed.

7.0 REFERENCES / FURTHER READINGS

Barbara K. Kaye, Norman J. Medoff (2001) The World Wide Web: A Mass Communication Perspective, Mayfield Publishing Company, USA

Barrie Sosinsky, Valda Hilley, (2004) Programming the Web: An Introduction, McGraw-Hill, USA

Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Dave Mercer, Schaum's Outline of HTML, Introduction to Web Page Design and Development McGraw-Hill, USA, 2002

Deitel P. J., and Deitel H.M (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

June Cohen (2003), The Unusually Useful Web Book, New Rider, USA

Marc D. Miller, Thomas C. Padgett (2003) Web Design Using Macromedia Dreamweaver, McGraw-Hill/ Irwin, USA

Peter Weverka (2001) Instant Web Pages, Sybex Inc, USA

Raymond Greenlaw and Ellen Hepp (2001), In-Line/On-Line Fundamentals of the Internet and World Wide Web, McGraw-Hill Learning Center, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

<http://www.daaq.net/old/javascript/index.php?page=js+iteratives&parent=js+statements>

UNIT 4: EVENTS AND EVENT HANDLERS IN JAVASCRIPT

CONTENTS

- 1.0 INTRODUCTION
- 2.0 OBJECTIVES
- 3.0 MAIN CONTENT
 - 3.1 JavaScript Popup Boxes
 - 3.2 JavaScript Functions
 - 3.3 JavaScript Events
 - 3.4 Events Handlers
- 4.0 CONCLUSION
- 5.0 SUMMARY
- 6.0 TUTOR MARKED ASSIGNMENT
- 7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

The word event as used in relation to computer programming usually signifies some sort of action or occurrence. As will be further discussed in this unit, an event refers to a repositioning of the mouse cursor, a mouse click, the filling of a form, or the pressing of the enter key. JavaScript lets you react to these events by specifying the relevant attribute in the object's HTML tag called an event handler. To use an event handler, it has to be included in the HTML tag. Most times, a function is created to handle an event. A function is lines of JavaScript code that perform some action or action(s).

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to implement JavaScript Popup Boxes
- to explain the meaning of event and events handler s
- to use JavaScript Functions
- to use JavaScript to implement events and event handlers

3.0 MAIN CONTENT

3.1 JavaScript Popup Boxes

Popup boxes are used to display a message, along with an "OK" button. Depending on the popup box, it might also have a "Cancel" button, and you might also be prompted to enter some text. JavaScript has three different types of popup box available for you to use. They are Alert box, Confirm box, and Prompt box.

a.) Alert Box

An alert box is often used if you want to make sure information comes through to the user. When an alert box pops up, the user will have to click "OK" to proceed.

Syntax

```
alert("sometext");
```

Example 1

```
<html>
<head>
<script type="text/javascript">
function show_confirm()
{
var r=confirm("Press a button");
if (r==true)
{
    alert("You pressed OK!");
}
else
{
    alert("You pressed Cancel!");
}
}
</script>
</head>
<body>

<input type="button" onclick="show_confirm( )" value="Show confirm box" />

</body>
</html>
```

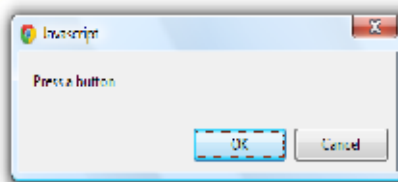


Figure 1: Alert

b.) Confirm Box

A confirm box is often used if you want the user to verify or accept something. When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed. If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Syntax

```
confirm("sometext");
```

Example 2

```
<html>
<head>
<script type="text/javascript">
function show_confirm()
{
var r=confirm("Press a button");
if (r==true)
{
    alert("You pressed OK!");
}
else
{
    alert("You pressed Cancel!");
}
}
</script>
</head>
<body>
<input type="button" onclick="show_confirm( )" value="Show confirm box" />
</body>
</html>
```

c.) Prompt Box

A prompt box is often used if you want the user to input a value before entering a page. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax

```
prompt("sometext","defaultvalue");
```

Example 3

```
<html>
<head>
<script type="text/javascript">
function show_prompt()
{
var name=prompt("Please enter your name","Myname");
if (name!=null && name!="")
{
document.write("Hello " + name + "! You are Welcome!");
}
}
</script>
</head>
<body>
<input type="button" onclick="show_prompt()" value="Show prompt box" />
</body>
</html>
```

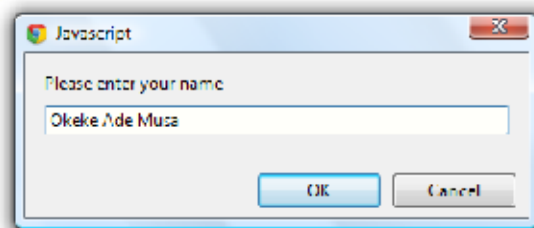


Figure 1: Prompt

3.2 JavaScript Functions

A function contains codes that will be executed by an event or by a call to the function. You may call a function from anywhere within a page (or even from other pages if the function is embedded in an external .js file). Functions can be defined both in the <head> and in the <body> section of a document. However, to assure that a function is read/loaded by the browser before it is called, it is wise to put functions in the <head> section.

How to Define a Function

Syntax

```
function functionname(var1,var2,...,varX)
{
some code
}
```

The parameters var1, var2, etc. are variables or values passed into the function. The { and the } defines the start and end of the function.

Note: A function with no parameters must include the parentheses () after the function name.

Note the word function is in lower case and when a call is made it has to be spelt correctly.

Example 4

```
<html>
<head>
<script type="text/javascript">
function nounmessage()
{
alert("Welcome to National Open University of Nigeria!");
}
</script>
</head>
<body>
<form>
<input type="button" value="Click me!" onclick=" nounmessage ()" />
</form>
</body>
</html>
```

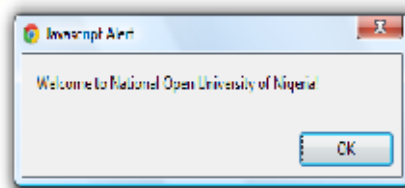


Figure 3: Welcome

If the line: alert("Welcome to National Open University of Nigeria!!") in the example above had not been put within a function, it would have been executed as soon as the page was loaded. Now, the script is not executed before a user hits the input button. The function nounmessage () will be executed if the input button is clicked.

The return Statement

The return statement is used to specify the value that is returned from the function. So, functions that are going to return a value must use the return statement.

The example below returns the area of a rectangle i.e length * breadth

Example 5

```
<html>
<head>
<script type="text/javascript">
function area (length ,breadth)
{
return length*breadth;
}
</script>
</head>
<body>
<script type="text/javascript">
document.write(area (10,15));
</script>
</body>
</html>
```

3.3 JavaScript Events

JavaScript programs do not have to be executed in sequence. We can make web pages more interactive by using Events. These are actions that can be detected by JavaScript. A wide variety of events enable your scripts to respond to the mouse, the keyboard, and other circumstances. Examples of Events are:

- A web page or an image loading
- Mouse click
- Mouse over a hot spot on the web page
- Selecting an input field in an HTML form
- Submitting an HTML form
- A keystroke

The script that you use to detect and respond to an event is called an event handler. Event handlers are among the most powerful features of JavaScript.

3.4 Events Handlers

In JavaScript/HTML, an event handler allows you to attach your JavaScript to your HTML elements.

Event handlers allow your web page to detect when a given "event" has occurred, so that it can run some JavaScript code. In your code, an event handler is simply a special attribute that you

add to your HTML element. For example, to run some JavaScript when the user clicks on an element, you add the onClick attribute to the element. Some of these Event handles have been used in the examples above.

More Examples of Event Handlers:

Event	Description
onclick:	Use this to invoke JavaScript upon clicking (a link, or form boxes)
onload:	Use this to invoke JavaScript after the page or an image has finished loading.
onmouseover:	Use this to invoke JavaScript if the mouse passes by some link
onmouseout:	Use this to invoke JavaScript if the mouse goes pass some link
onunload:	Use this to invoke JavaScript right after someone leaves this page.
onSubmit	The onSubmit event is used to validate ALL form fields before submitting it.
onFocus, onBlur and onChange	The onFocus, onBlur and onChange events are often used in combination with validation of form fields.

4.0 CONCLUSION

Event such as the onclick and onsubmit events can be used to trigger scripts. JavaScript events which allow scripts to respond to users' interaction and modify the pages accordingly have been discussed in this unit.

5.0 SUMMARY

One very simple response to an event is to display a dialog box. JavaScript provides three types of dialog boxes: alert box, confirmation box, and prompt box. Events allow scripts to respond to a user who is moving the mouse, entering form data or pressing keys. Events and events handling help make web application more responsive, dynamic and interactive.

6.0 TUTOR MARKED ASSIGNMENT

- Name three JavaScript event handler and describe how they are used. Create a Web page that incorporates them.
- What are some practical uses of alert boxes?

7.0 REFERENCES / FURTHER READINGS

Barbara K. Kaye, Norman J. Medoff (2001) The World Wide Web: A Mass Communication Perspective, Mayfield Publishing Company, USA

Peter Weverka (2001) Instant Web Pages, Sybex Inc, USA

Raymond Greenlaw and Ellen Hepp (2001), In-Line/On-Line Fundamentals of the Internet and World Wide Web, McGraw-Hill Learning Center, USA

Marc D. Miller, Thomas C. Padgett (2003) Web Design Using Macromedia Dreamweaver, McGraw-Hill/ Irwin, USA

Dave Mercer, Schaum's Outline of HTML, Introduction to Web Page Design and Development McGraw-Hill, USA, 2002

June Cohen (2003), The Unusually Useful Web Book, New Rider, USA

Behrouz A Forouzan (2003), Data Communications and Networking, Third Edition, International Edition, McGraw-Hill/Osborne, N.Y, USA

Barrie Sosinsky, Valda Hilley, (2004) Programming the Web: An Introduction, McGraw-Hill, USA

P. J. Deitel, and H.M Deitel (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

MODULE FOUR

UNIT 1: OVERVIEW OF XML

CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES

3.0 MAIN CONTENT

3.1 XML Basics

3.2 Creating and Modifying XML Documents

3.3 Processing XML documents

3.4 Validating XML Documents

3.5 Formatting and Manipulating XML Documents

3.6 Viewing an XML Document in Web Browser

4.0 CONCLUSION

5.0 SUMMARY

6.0 TUTOR MARKED ASSIGNMENT

7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. It was developed in 1996 by the World Wide Web Consortium's (W3C's) XML working group. As a widely supported open technology for describing, storing and sharing data, Web application use XML extensively and current versions of most browsers provide huge support for the language. The primary design goals of XML emphasize simplicity, generality, and usability over the Internet. Today, XML has grown to become the most common tool for data transmissions between all sorts of applications. It is a textual data format with strong support via Unicode for the world's languages. Though XML focuses on documents, it is widely used for the representation of arbitrary data structures e.g. Web services. There are many applications programming interfaces (APIs) used to process XML data and several schema systems exist to aid in the definition of XML-based languages. We shall learn more about XML in this unit.

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to explain the meaning of XML
- to explain how to create and modify XML document
- to explain how to process and validate XML document
- to explain how to view XML documents in a Web browser

3.1 XML Basics

XML permits document authors to create markup (i.e. text-based notation for describing data) for virtually any type of information. This enables document authors to create entirely new markup languages for describing any type of data, such as mathematical formula, software

configuration instructions, chemical molecular structures, music, new recipes and financial reports. XML describes data in a way that both human beings and computer can understand. XML is not a replacement for HTML. HTML is about displaying information, while XML is about carrying information. XML uses tags to structure data. The tags are not predefined-every developer is expected to define his/her tags. XML is designed to be self-descriptive. Tags are markup construct that begins with "<" and ends with ">". Tags come in three flavors: *start-tags, for example <section>, end-tags, for example </section>, and empty-element tags, for example <line-break />.* An element's start and end tags enclose text that represents a piece of data. Every XML document must have exactly one root element that contains all the elements. XML documents may begin by declaring some information about themselves, as in the following example.

```
<?xml version="1.0" encoding=" ISO-8859-1" ?>
```

Now let us Let as take a look at these simple XML code below:

Example 1:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MyPersonalDetails>
  <FullName>
    <FirstName>Musa </FirstName>
    <LastName>Okeke</LastName>
  </FullName>
  <BirthDate>
    <Month>June</Month>
    <Date>17</Date>
    <Year>1970</Year>
  </BirthDate>
  <MailingAddress>
    <University>National Open University of Nigeria </University>
    <School>Science and Technology</School>
    <Programme>Computer Science</Programme>
  </MailingAddress>
</MyPersonalDetails>
```

From the codes above, XML did nothing at all. It is just information wrapped in tags. Someone must write a piece of software to send, receive or display it. The first line of code tells the version and character encoding being used by this XML document. The second line of code tells what kind of information or XML document. The XML applications that will use the codes in example 1 will look at the root or parent tag in the XML document. Here it is <MyPersonalDetails >, which is not defined by XML. XML allows authors to create their own XML tag to be used in each document. XML, like any other languages, is capable of having two or more child tags or commonly known as nested tags. The <FullName> tag has three child tags, so on and so forth. Also, XML tags are not case sensitive. Meaning we can declare <MyPersonalDetails > opening tag with a closing tag of </myPersonalDetails >.

3.2 Creating and Modifying XML Documents

XML allows you to describe data precisely in a well-formed format. XML documents are highly portable. Any text editor such as Notepad or software that supports ASCII/Unicode characters can open XML documents for viewing and editing. An XML document is created by typing XML codes into a text editor and then saving the document with a filename and a .xml extension. Most Web browsers can display XML documents in a formatted manner that shows the XML's structure.

3.3 Processing XML documents

To process an XML document you would require an XML parser (or XML processor). A parser is software that checks that the document follows the syntax rules specified by the W3C's XML recommendation and makes the document's data available to an application. A parser would, for example, check an XML document to ensure that there is a single root element, a start tag for each element, and properly nested tags (i.e. the end tag for a nested element must appear before the end tag of the enclosing element). Furthermore, XML is case sensitive, so the proper capitalization must be used in elements as in Example 1. A document that conforms to this syntax is said to be a well-formed XML document and is syntactically correct. If an XML parser can process an XML document successfully, that XML document is well-formed. Parsers can provide access to XML-encoded data in a well-formed document only. Often XML parsers are built into software or available for download over the Internet. Examples of parsers include Microsoft XML Core Services (MSXML), Xerces Expat, etc.

3.4 Validating XML Documents

In addition to being well-formed, an XML document may be valid. This means that it contains a reference to a Document Type Definition (DTD), and that its elements and attributes are declared in that DTD and follow the grammatical rules for them that the DTD specifies. A DTD is an example of a schema or grammar. Since the initial publication of XML 1.0, there has been substantial work in the area of schema languages for XML. Such schema languages typically constrain the set of elements that may be used in a document, which attributes may be applied to them, the order in which they may appear, and the allowable parent/child relationships. When an XML document references a DTD or a schema, some parsers (called validating parsers) can read the DTD/Schema and check that the XML conforms to the DTD/Schema; the XML document is valid. For example, if in Figure 2.1 we were referencing a DTD that specifies that `<MonthData>` must have `Month`, `Date`, and `Year` elements, then the exclusion of a `Year` element would invalidate the XML document `detail2.xml`. However, the XML document would still be well-formed, because it follows proper XML syntax (i.e. it has one root element, each element has a start tag and an end tag, and the elements are nested properly). By definition, a valid XML document is well-formed. Parsers that cannot check for document conformity against DTDs/schemas are non-validating parsers—they determine only whether an XML document is well-formed, not whether it is valid. Schemas are XML documents themselves, whereas DTDs are not. XML processors are classified as validating or non-validating depending on whether or not they check XML documents for validity. A processor that discovers a validity error must be able to report it, but may continue normal processing.

3.5 Formatting and Manipulating XML Documents

XML document can be manipulated to appear differently on several devices. For example, the way XML document renders on Personal Digital Assistants (PDAs) is different from Desktop computers. Most XML documents contain only data. They do not include formatting instructions, so applications that process XML documents must look for how to process, manipulate or display the data. Extensible Stylesheet Language (XSL) can be used to specify rendering instructions for different platforms. XML-processing programs can also search, sort and manipulate XML data using XSL. Other popular XML-related technologies are: XPath XML Path Language (XPath) which is used for accessing parts of an XML document, XSL Formatting Objects (XSL-FO) which is a XML vocabulary used to describe document formatting, and XSL Transformations-language (XSLT) used for transforming XML documents into other documents.

3.6 Viewing an XML Document in Web Browser

Example 1 shows a simple listing of a text file for detail2.xml. This document does not contain formatting information for the detail2.xml. This is because XML is a tool for describing the structure, storage and transferring of data across disparate format/sources. Formatting and displaying data from an XML document is achieved in different ways within specific application platform. For instance, when the user loads detail2.xml in the Internet Explorer, MSXML (Microsoft XML Core Services) or Firefox, it will be parsed and display the document data. Each browser has a built-in style sheet to format the data. Note that the resulting format of the data in Example 1 is similar to the format of the listing in Figure 2.1. The XML document will be displayed with color-coded root and child elements. A plus (+) or minus sign (-) to the left of the elements can be clicked to expand or collapse the element structure. To view the raw XML source (without the + and - signs), select "View Page Source" or "View Source" from the browser menu. Although these symbols are not part of the XML document, both browsers place them next to every container element. A minus sign indicates that the browser is displaying the container element child element. Clicking the minus sign next to an element collapses that element. (i.e. causes the browser to hide the container element's children) and replace the minus sign with a plus). Conversely, clicking the plus sign next to an element expands the elements (i.e. causes the browser to display the container element's children and replaces the plus sign with a minus sign). This behavior is similar to viewing the directory structure on your system in Windows Explorer or another similar directory viewer. In fact a directory structure often is modeled as a series of tree structure in which the root of the tree represents a disk drive e.g. C: and nodes in the tree represent directories. Parsers often store XML data as tree structure to facilitate efficient manipulation as discussed.

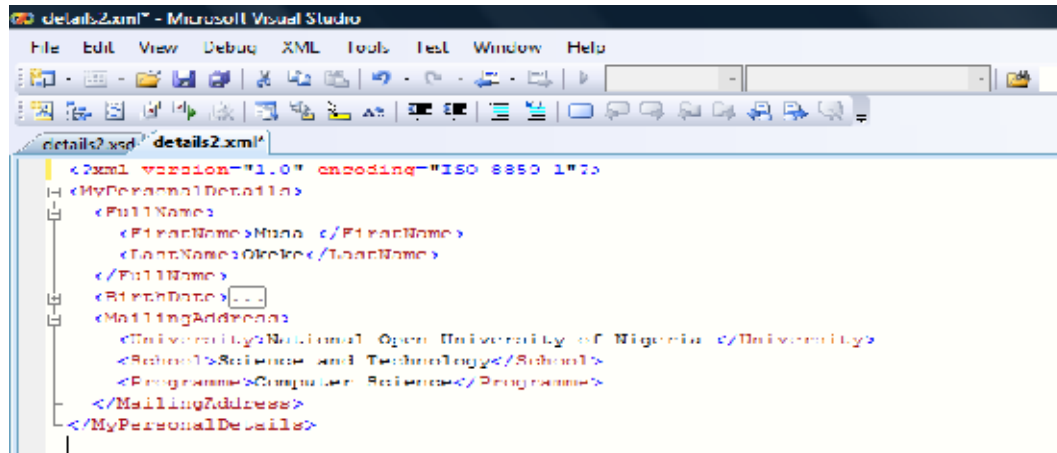


Figure 2.1: View of XML file in a Browser

4.0 CONCLUSION

Within the last two decades of the introduction of XML, it has been used to create hundreds of languages which include XHTML, WSDL for describing available web services, WAP and WML as markup languages for handheld devices, RSS languages for news feeds, RDF and OWL for describing resources and ontology, SMIL for describing multimedia for the web etc. In addition, XML-based formats have become the default for most office-productivity tools, including Microsoft Office (Office Open XML) and Apple's iWork

5.0 SUMMARY

XML describes data in a way that both human beings and computer can understand. It enhances the storage and exchange of data amongst disparate computer systems. In this unit we covered how to create, modify, validate, format, process and view XML documents in a browser.

6.0 TUTOR MARKED ASSIGNMENT

- a) Discuss the importance of XML technologies
- b) State which of the following statements is true(T) or false(F)
 - i) XML can be used to create other markup languages(T/F)
 - ii) Parser are used to check the syntax of an XML document (T/F)
 - iii) XML document is not case sensitive (T/F)
 - iv) All XML start tags must have corresponding end tags (T/F)

7.0 REFERENCES / FURTHER READINGS

Deitel, P. J. and Deitel H.M (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

UNIT 2: SEARCH ENGINES & TOOLS

CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES

3.0 MAIN CONTENTS

3.1.1 Directories

3.1.2 Search Engines

3.2 Components of a Search Engine

3.3 Listing in a Search Engine and Search Index

4.0 CONCLUSION

5.0 SUMMARY

6.0 TUTOR MARKED ASSIGNMENT

7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

Quite a number of collections of search tools are available today that allow users to find information on the Web quickly and easily. Two basic approaches have evolved in response to the need to organize to locate information on the World Wide Web. These are directories and search engines. A directory offers a hierarchy representation of hyperlinks to Web pages and presentation broken down into topics and subtopics. On the other hand, a search engine is a set of programs that is used to search for information within a specific realm and collate that information in a database. Although search engine is really a general class of programs, the term is often used to specifically describe Internet search engine like Google, Alta Vista and Excite that enable users to search for documents on the World Wide Web, FTP servers and USENET newsgroups. Search engines can also be devised for offline content, such as a library catalogue, the contents of a personal hard drive, or a catalogue of museum collections. Generally search engines help people to organize and display information in a way which makes it readily accessible.

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to explain the meaning of Search Tools
- to describe the components of a Search Engine
- to explain how search engines works
- to discuss how to popularize an a site
-

3.0 Main Contents

3.1 Search Tools

A search tools is software that enables a user to quickly and easily gain access to information. The collection of search tools is constantly evolving with new ones coming on the scene and others disappearing. In this unit we shall consider two basic approaches that have evolved in response to the need to organize and locate information on the World Wide Web i.e directories

and search engines. Both approaches allow information about Web pages that is contained in some database that already been created, either manually or using special programs that search the Web pages to be assess quickly and easily. A request for information is answered by the search tool retrieving the information from its already-constructed database of indexed Web details. Other definitions that relate to searching information on the Web are as follows:

Search Terminology

Search tool – This refers to any mechanism for locating information on the Web. Examples include search or metasearch engine, directory, etc.

Metasearch engine – This refers to an all-in-one search engine that performs a search by calling on more than one other search engine to do the actual work.

Query – This refers to the information entered into a form on a search engine's Web page that describes the information being sought.

Query Syntax – This terms is used to describe, the set of rules describing what constitute a legal query on some search engines, special symbols may be used in a query

Query Semantic – This term is used to describe a set of rules that defines the meaning of a query

Hit – This refers to a URL that a search engine returns in response to a query

Match – This is a synonym for hit

Relevancy score – This refers to a value that indicates how close a match, a URL was to a query; usually expressed as a value from 1 to 100, with the higher score meaning more relevant.

3.1.1 Directories

The first method of finding and organizing Web information is stated earlier is the **directory approach**. A directory offers a hierarchy representation of hyperlinks to Web pages and presentation broken down into topics and subtopics. The hierarchy can descend many levels. The specific number of levels is determined by the taxonomy of topics.

Examples of popular general directories

www.looksmart.com

www.lycos.com

www.dmoz.com

www.yahoo.com

3.1.2 Search Engines

The second approach to organizing information and locating information on the Web is a search engine, which is a computer program that does the following:

1. Allows a user to submit a form containing a query that consist of a word or phrase describing the specific information of interest to be located from the Web.
2. Searches it database to try to match your query
3. Collate and returns a list of clickable URLs containing presentations that match the users query; the list is usually ordered with the better matches appearing at a the top
4. Permits a user to revise and resubmit a query

A recent survey ranking the market share of web search engine carried out by Net Marketshare in December 2010, showed

Google is 84.65%,
Yahoo is 6.69%,
Baidu is 3.39%,
Bing is 3.29% and
Other is 1.98%.

3.2 Components of a Search Engine

Search engines have the following components

- a) User Interface
- b) Databases
- c) Robot or Spider Software

a) User Interface:

The users interface is a mechanism by which users submit queries to the search engine by typing a keyword or phrases to search into the text box. When the form is submitted, the data typed into the text box is sent to a server-side script that searches the database using the keywords entered. Afterwards search results are displayed in the browser containing a list of information, such as the URLs for Web pages that meet the users' criteria. This result set is formatted with a link to each page along with additional information that might include the page title, a brief description, the first few lines of text, or the size of the page and a relevancy score for each hit. This way, the user is able to make an informed choice as to which hyperlinks to follow. Hyperlinks to help files are usually displayed prominently, and advertisement should not hinder a reader's use of the search engine. The order in which pages are displayed may depend on paid advertisement, alphabetical order, and link popularity. Each search engine has its own policy for ordering the search results. The policies can change over time.

b.) Database

A database is a collection of information organized so that its contents can easily be accessed, managed and updated. Databases management systems (DBMSs) such as Oracle, Microsoft SQL Server, Informix, MySQL or IBM DB2 are used to configure and manage the database. The

databases associated with search engines are extremely large indexed pages that require a highly efficient search strategy to retrieve information from them. Computer scientists have spent years developing efficient several searching and sorting strategies which are implemented in the search. The information displayed as results of your search is usually from the database accessed by the search engine site. Some search engines, such as AOL and Netscape use a database by Google.

c.) Robot

A robot (sometimes called a spider) is a program that automatically traverses the hypertext structure of the Web by retrieving a Web page document and following the hyperlinks on the page. It moves like a robot spider on the Web, accessing and documenting Web pages. It requests pages from a website in the same way as Microsoft Explorer, or Firefox and any other browser do it. Spider doesn't collect images or formatting details. It is only interested in text and links and the URL from which they come. The spider categorizes the pages and stores information about the Web site and the Web pages in a database. Various robots may differently, but in general, they access and may store important information on web pages such as: title, meta tag keyword, meta tag description, and some of the text on the page (usually either the first few sentences of the text contained in the heading tags). For multimedia elements in web pages to be index, the =alt' tag should be used in order to have values in the search engines. The spider software works in conjunction with the index software. This uses the information collected by the spider. The spider takes the information it has gathered about a web page and sends it to the index software where it is analyzed and stored. The index makes sense of the mass of text, links and URLs using an algorithm which refers to a complex mathematical formula that indexes the words, the pairs of words and so on. The algorithm analyzes the pages and links for word combinations to determine what the web pages are all about i.e what topics are being. Then, scores are assigned that allow the search engine to measure how relevant important the web pages (and URLs) might be to the user or visitor. Major search engines such as Google, Yahoo or Bing use proprietary algorithm for scoring.

3.3 Listing in a Search Engine and Search Index

The components of a search engine (robot, database and search form) work together to obtain information about Web pages, store information about Web pages, and provide a graphical user interface to facilitate searching for and displaying a list of Web pages relevant to given words. In recent times, search engines as become one of the top methods used to drive traffic to ecommerce sites. Though very effective, it is not always easy to get listed in a search engine or search directory. These days there is a trend away from free listing in search engines. Current trends entail paying for listing consideration in a search engine or directory. These approaches include an express submit or express inclusion, paying for preferential placement in search engine displays (called sponsoring or advertising), and paying each time a visitor clicks the search engine's link to your site. Yahoo and Google use the terms Calls its Sponsor Results and Google AdWords respectively. In these programs, payment is made when the site is submitted for review. If accepted, the site has a listing usually at the top or right margin of the search

results. In addition to the initial fee, the Web site owners must pay each time a visitor clicks on the search engine link to their site-this is called a cost-per-click (CPC).

4.0 CONCLUSION

A web search engine is designed to search for information on the World Wide Web, FTP servers USENET newsgroup, etc. The search results which may consist of web pages, images, information and other types of files are generally presented in a list of results and are often called hits. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained by human editors, search engines operate algorithmically or are a mixture of algorithmic and human input.

5.0 SUMMARY

Search engines use automated software programs to survey the Web and build their databases. Web documents are retrieved by these programs and analyzed. Data collected from each web page are then added to the search engine index. Each search engine uses a proprietary algorithm to create its indices such that, ideally, only meaningful results are returned for each query. The best URLs are then returned to the user as hits, ranked in order with the best results depending on the algorithm used by the search engine at the top.

6.0 TUTOR MARKED ASSIGNMENTS

- 1) Describe the components of a search engine
- 2) Describe how Google search engine works?
- 3) What is the meaning of Google AdWords?

7.0 REFERENCES / FURTHER READINGS

Deitel, P.J. and Deitel H.M (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

Robert W. Sebesta (2009), Programming the World Wide Web, Pearson Addition-Wesley, New Jersey, USA

UNIT 3: THE FUTURE WEB, TECHNOLOGIES AND DEVELOPMENT TOOLS

CONTENTS

- 1.0 INTRODUCTION
- 2.0 OBJECTIVES
- 3.0 MAIN CONTENT
 - 3.1 Web Services
 - 3.2 Cloud Computing
 - 3.3 Blogs
 - 3.5 Podcasts
 - 3.7 Microformat
 - 3.8 Resources Description Framework (RDF)
 - 3.9 Ontologies
 - 3.10 Application Programming Interface (APIs)
 - 3.11 Mashups
 - 3.12 Widgets and Gadgets
 - 3.13 Web 2.0
 - 3.14 Web 2.0 Tools
- 4.0 CONCLUSION
- 5.0 SUMMARY
- 6.0 TUTOR MARKED ASSIGNMENT
- 7.0 REFERENCES / FURTHER READINGS

1.0 INTRODUCTION

Advances in Internet technology has led to the release of several tools for Web development. Many of the tools are easy to use and made available to the public free of charge to development. A popular example is the LAMP (Linux, Apache, MySQL, PHP) stack, which is usually distributed free of charge. The availability of free tools has greatly influence the rate at which many people around the globe setup new Web sites daily. Easy to use software for Web development include amongst others: Adobe Dreamweaver, Netbeans, WebDev, or Microsoft Expression Studio, Adobe Flex etc. By using these software virtually anyone can develop a Web page in a matter of minutes. Knowledge of Hypertext Markup Language (HTML) or other programming languages is not usually required, but recommended for professional results. Newer generation of web development tools use the strong growth in LAMP, Java Platform, Enterprise Edition technologies and Microsoft .NET technologies to provide the Web as a way to run applications online. Web developers now help to deliver applications as Web services which were traditionally only available as applications on a desk based computer. Thus, instead of running executable code on a local computer, users can now are interact with online applications to create new contents. This has enabled new methods in communication and allowed for many opportunities to decentralize information and media distribution. In this unit we shall discuss other technologies, models and tools that enhance easy development of Web applications.

2.0 OBJECTIVES

At the end of this unit, the students will be able:

- to describe the tools used for Web applications
- to explain the meaning of Web 2.0 Technologies

- to classify and use Web development tools
- to learn current and emerging trends in Web application development

3.0 Main Contents

3.1 Web Services:

There is no need reinventing the wheels with every new project. With web services, developers can use existing company's software solutions to create feature-rich applications. A Web service is a self-describing, self-contained application that provides some business functionality through an Internet connection. For example, an organization could create a Web service to facilitate information exchange with its partner or vendor. The Universal Discovery, Description, and Integration (UDDI) standard, is backed by a number of technology companies, including IBM, Microsoft, and Sun Microsystems. Essentially, UDDI provides a method of describing a service, involving a service, and locating available services. Microsoft's .NET platform supports Web services. The incorporation of web services into new programs allows the speedy development of new applications. A Web service is a method of communication between two electronic devices over a network. It has an interface described in a machine-processable format (specifically Web Services Description Language WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Two major classes of Web services i.e. REST-compliant Web services, in which the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of "stateless" operations; and arbitrary Web services, in which the service may expose an arbitrary set of operations can be easily identified. Web services use Extensible Markup Language (XML) messages that follow the SOAP standard and have been popular with traditional enterprises. In such systems, there is often a machine-readable description of the operations offered by the service written in the Web Services Description Language (WSDL). The latter is not a requirement of a SOAP endpoint, but it is a prerequisite for automated client-side code generation in many Java and .NET SOAP frameworks (frameworks such as Apache Axis2, Apache CXF, and Spring being notable exceptions). Some industry organizations, such as the WS-I, mandate both SOAP and WSDL in their definition of a Web service.

3.2 Cloud Computing

Cloud computing refers to the use and access of multiple server-based computational resources via a digital network (WAN, Internet connection using the World Wide Web, etc.). Cloud users may access the server resources using a computer, netbook, pad computer, smart phone, PDA, or other devices. In cloud computing, applications are provided and managed by the cloud server and data is also stored remotely in the cloud configuration. Users do not download and install applications on their own device or computer; all processing and storage is maintained by the cloud server. The on-line services are usually offered by a cloud provider or by a private organization. Before the advent of cloud computing, tasks such as using word processing would not be possible without the installation of application software on a user's computer. A user would need to purchase a license for each application from a software vendor and obtained the right to install the application on one computer system. As computer technologies advanced, local area networks (LAN) and more networking capabilities, the client-server model of

computing were born, where server computers with enhanced capabilities and large storage devices could be used to host application services and data for a large workgroup. In a client-server computing environment, a network-friendly client version of the application was required on client computers which utilized the client system's resources (memory and CPU for processing), even though resultant application data files (such as word processing documents) were stored centrally on the data servers. In this case, multiple user licenses of an application were purchased for use by many users on a network. Cloud computing differs from the classic client-server model discussed in Module 1 of this course material, by providing applications from a server that are executed and managed by a client's web browser, with no installed client version of an application required. Cloud computing provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. You may compare this scenario with the concept drawn from the electricity grid, wherein end-users consume power without needing to understand the component devices or infrastructure required to provide the service. Centralization gives cloud service providers complete control over the versions of the browser-based applications provided to clients, which removes the need for version upgrades or license management on individual client computing devices. The term Software as a service (SaaS) is sometimes used to describe application programs offered through cloud computing. Cloud computing describes a new supplement, consumption, and delivery model for IT services based on Internet protocols, and it typically involves provisioning of dynamically scalable and often virtualized resources. It is a by-product and consequence of the ease-of-access to remote computing sites provided by the Internet. This may take the form of web-based tools or applications that users can access and use through a web browser as if they were installed locally on their own computers. Most cloud computing infrastructures consist of services delivered through shared data-centres. The Cloud may appear as a single point of access for consumers' computing needs; notable examples include the iTunes Store and the iOS App Store. Commercial offerings may be required to meet service level agreements (SLAs), but specific terms are less often negotiated by smaller companies.

3.3 Blogs

A blog is a blend of the term web log. It is a type of website or part of a website. Many blogs provide commentary or news on a particular subject; others function as more personal online diaries. A typical blog combines text, images, and links to other blogs, Web pages, and other media related to its topic. The ability of readers to leave comments in an interactive format is an important part of many blogs. Most blogs are primarily textual, although some focus on art (art blog), photographs (photoblog), videos (video blogging), music (MP3 blog), and audio (podcasting). Microblogging is another type of blogging, featuring very short posts. Most blogs are interactive, allowing visitors to leave comments and even message each other via widgets on the blogs and it is this interactivity that distinguishes them from other static websites. Entries are commonly displayed in reverse-chronological order. Many blogs are hosted at blog communities such as <http://blogspot.com>

3.4 RSS

Really Simple Syndication or Rich Site Summary (RSS) is commonly used to create newsfeed from blog postings and other Web sites. The RSS feeds contain a summary of new items posted to the site. Web feeds benefit publishers by letting them syndicate content automatically. They benefit readers who want to subscribe to timely updates from favored websites or to aggregate feeds from many sites into one place. RSS feeds can be read using software called an "RSS", "feed reader", or "aggregator", which can be web-based, desktop-based, or mobile-device-based. Some browser, such as Firefox, Safari, and Internet 7 can display RSS feeds. A standardized XML file format allows the information to be published once and viewed by many different programs. The user subscribes to a feed by entering into the reader the feed's URI or by clicking a feed icon in a web browser that initiates the subscription process. The RSS reader checks the user's subscribed feeds regularly for new work, downloads any updates that it finds, and provides a user interface to monitor and read the feeds. RSS allows users to avoid manually inspecting all of the websites they are interested in, and instead subscribe to websites such that all new content is pushed onto their browsers when it becomes available. By providing up-to-date, linkable content for anyone to use, RSS enables website developers to draw more traffic. It also allows users to get news and information from many sources easily and reduces developers time. RSS simplifies importing information from portals, weblogs and news sites. Any piece of information can be syndicated via RSS, not just news.

3.5 Podcasts

Podcasts are typically audio files, delivered by an RSS feed on the Web. They may also be made available by recording an MP3 file and providing a link on a Web page. They usually would take the format of an audio blog, interview or radio show. These files can be saved to your computer or to an MP3 player (such as iPod) for later listening.

3.6 Wiki

A wiki is a Web site that can be updated immediately at any time by visitors using a simple form on a Web page. Some wikis are intended for a small group of people such as the members of an organization. The most powerful wiki is Wikipedia (<http://Wikipedia.org>), which is an online encyclopaedia which can be updated by any registered user at anytime. Wiki is a form of social software in action where visitors sharing their collective knowledge can create a resource freely used by all. Though there have been isolated cases of practical jokes and occasionally inaccurate information posted at Wikipedia, the information and resources provided is still good enough as starting point when exploring a topic.

3.7 Microformat

Microformat is a standard format for representing information aggregate that can be understood by computers, enabling better search results and new types of applications. Some people consider the web as containing loose information while others see logical aggregates, business cards, resume, events, etc. Thus there is the need to organize information in a standard way. Microformat standard encourage site to organize their information such that its increases interoperability. For example, if you want to create an event or an events calendar, you could user the hCalalender microformat. Some other microformat are the adr for address information, hresume for resume and xfolk for collections of bookmarks.

3.8 Resources Description Framework (RDF)

The Resource Description Framework (RDF), developed by the World Wide Web consortium (W3C) is one way of making the Web more meaningful. It is based on XML and used to describe content in a way that is understood by computers. RDF helps connect isolated databases across the web with consistent semantics. The structure of any expression in RDF is a collection of triples. RDF triples consist of two pieces of information (subject and object) and linking fact (predicate).

3.9 Ontologies

Items on the Web need to be organized in such a way that meaning can be easily derived from them. Ontologies are ways of organizing and describing related items, and are used to represent semantics. It is another means of cataloguing Internet content in a way that can be understood by the computers. RDF and OWL (Web Ontology Language), also designed for formatting ontologies. However, OWL for formatting ontologies in XML, extend beyond the basic semantics of RDF ontologies to enable even deeper machine understanding of content.

3.10 Application Programming Interface (APIs)

Application Programming Interface (APIs) provides application with access to external services and databases. For example, a traditional programming API, like the Sun's Java API, allows programmers to use already-written methods and functions in their programs. Also APIs exists for Web services have APIs that permit their functionality and information to be shared or used across the internet. Much of the value of a site is in its databases. Many major Web companies (e.g, eBay, Amazon, Google, Yahoo! and Flickr) provide APIs to encourage use of their services and data in the development of mashups, widgets or gadgets.

3.11 Mashups

Mashups is combine contents or functionality from existing web services, websites and RSS feeds or other solutions to serve a new purpose. For example, a developer can mashup Google Maps with a tourist site or a real estate site. Mashups with maps are particularly popular, as are mashup using RSS feeds created by using services such as Yahoo! Pipes. This is one tool that enables a developer to aggregate and manipulate many data sources. The use of APIs can save time and money. In fact some great mashups can be built a day. However, the mashup may rely on one or more third parties software. Thus, if the API provider experiences downtime, the mashup will be unavailable as well (unless the mashup is programmed to avoid sites that are down). It is recommended that a developer check the term of service for using a third party software for the purpose of mashuup. Also note that many API providers charge

3.12 Widgets and Gadgets

Widgets are commonly referred to as gadgets. They are mini applications designed to run either as stand alone or as add-on features in web pages. Widgets can be used to personalize Internet experience by displaying real-time weather conditions, viewing maps, receiving event reminder, providing easy access to search engines, aggregating RSS feeds, etc. The viability of web services, APIs and various tools make it easy to develop Widgets. Several catalogs of widgets exist online with the most all-inclusive being Widgipedia. This catalog provides an extensive widgets and gadgets for a variety of platform.

3.13 Web 2.0

The term Web 2.0 is associated with web applications that facilitate participatory information sharing, interoperability, user-centered design, and collaboration on the World Wide Web. A Web 2.0 site allows users to interact and collaborate with each other in a social media dialogue as creators (prosumers) of user-generated content in a virtual community, in contrast to websites where users (consumers) are limited to the passive viewing of content that was created for them. Examples of Web 2.0 include social networking sites, blogs, wikis, video sharing sites, hosted services, web applications, mashups and folksonomies. Web 2.0 websites allow users to do more than just retrieve information. By increasing what was already possible in Web 1.0, they provide the user with more user-interface, software and storage facilities, all through their browser. Users can provide the data that is on a Web 2.0 site and exercise some control over that data. These sites may have an "Architecture of participation" that encourages users to add value to the application as they use it. The Web 2.0 offers all users the same freedom to contribute.

3.14 Web 2.0 Tools

The client-side/web browser technologies used in Web 2.0 development are Asynchronous JavaScript and XML (Ajax), Adobe Flash and the Adobe Flex framework, and JavaScript/Ajax Dojo Toolkit, MooTools, jQuery, etc. Ajax programming uses JavaScript to upload and download new data from the web server without undergoing a full page reload. To allow users to continue to interact with the page, communications such as data requests going to the server are separated from data coming back to the page (asynchronously). Otherwise, the user would have to routinely wait for the data to come back before they can do anything else on that page, just as a user has to wait for a page to complete the reload. This also increases overall performance of the site, as the sending of requests can complete quicker independent of blocking and queuing required to send data back to the client. The data fetched by an Ajax request is typically in XML or JSON (JavaScript Object Notation) format which constitute the two widely used structured data formats. Since both of these formats are natively understood by JavaScript, a programmer can easily use them to transmit structured data in their web application. When this data is received via Ajax, the JavaScript program then uses the Document Object Model (DOM) to dynamically update the web page based on the new data, allowing for a rapid and interactive user experience. In short, using these techniques, Web designers can make their pages function like desktop applications. For example, Google Docs uses this technique to create a Web based word processor. Adobe Flex is another technology often used in Web 2.0 applications. Compared to JavaScript libraries like jQuery, Flex makes it easier for programmers to populate large data grids, charts, and other heavy user interactions. [Applications programmed in Flex, are compiled and displayed as Flash within the browser. Flash is capable of doing many things which were not possible pre-HTML5, the language used to construct web pages. Out of the many capabilities, of Flash, the most commonly used in Web 2.0 is its ability to play audio and video files. This has allowed for the creation of Web 2.0 sites where video media is seamlessly integrated with standard HTML. In addition to Flash and Ajax, JavaScript/Ajax frameworks have recently become a very popular means of creating Web 2.0 sites. At their core, these frameworks do not use technology any different from JavaScript, Ajax, and the DOM. What frameworks do is

smooth over inconsistencies between web browsers and extend the functionality available to developers. Many of them also come with customizable, prefabricated 'widgets' that accomplish such common tasks as picking a date from a calendar, displaying a data chart, or making a tabbed panel. On the server side, Web 2.0 uses many of the same technologies as Web 1.0. New languages such as PHP, Ruby, Perl, Python, JSP and ASP are used by developers to dynamically output data using information from files and databases. What has begun to change in Web 2.0 is the way this data is formatted. In the early days of the Internet, there was little need for different websites to communicate with each other and share data. In the new "participatory web", however, sharing data between sites has become an essential capability. To share its data with other sites, a website must be able to generate output in machine-readable formats such as XML (Atom, RSS, etc) and JSON. When a site's data is available in one of these formats, another website can use it to integrate a portion of that site's functionality into itself, linking the two together. This is one of the hallmarks of the philosophy behind the Web 2.0 movement.

4.0 CONCLUSION

The Internet is playing a great role in the delivery of contents to users all across the world. A lot of researches are going on every day to make it more accessible, available, interactive, meaningful and responsive to users' needs. Most of the information in this unit have been have been presented for you to keep up-to-date with current Internet and Web programming developments.

5.0 SUMMARY

In this unit we covered current and emerging topics in Web development which are aimed at making web contents easier to develop and having the responsiveness, look and feel of traditional desktop applications.

6.0 TUTOR MARKED ASSIGNMENT

Briefly discuss the meaning of the following terms

- a) Web services
- b) Blogs
- c) Cloud Computing
- d) Ontologies
- e) Widgets and Gadgets

7.0 REFERENCES / FURTHER READINGS

Deitel, P.J. and Deitel H.M (2008), Internet and World Wide Web: How to Program, Fourth Edition, Pearson Prentice Hall, New Jersey, USA

Terry Felke-Morris (2009), Web Development and Design Foundations with XHTML, Pearson International Edition USA

Robert W. Sebesta (2009), Programming the World Wide Web, Pearson Addition-Wesley, New Jersey, USA